

N 70 17465

NASA CR 107868

CONSTRUCTION OF CONVOLUTIONAL CODES
FOR SEQUENTIAL DECODING

Daniel J. Costello, Jr.

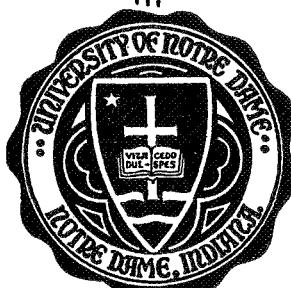
Technical Report EE-692

August, 1969

CASE FILE
COPY

Department of

ELECTRICAL ENGINEERING



UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

CONSTRUCTION OF CONVOLUTIONAL CODES
FOR SEQUENTIAL DECODING

Daniel J. Costello, Jr.

Technical Report EE-692

August, 1969

Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

This work was submitted to the Graduate School of the University of Notre Dame in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This work was supported by the National Aeronautics and Space Administration (NASA Grant 15-004-026) in liaison with the Flight Data Systems Branch of the Goddard Space Flight Center.

ACKNOWLEDGEMENTS

First of all, I must express my sincere and profound thanks to Professor James L. Massey, who not only spent many long hours providing comment on and insight into the work presented here, but whose excellent courses motivated me to do my research in this area. Working with Professor Massey has been a highly interesting, rewarding, and enjoyable educational experience. I am also indebted to Professor Ruey Wen Liu, who served as my advisor during Professor Massey's absence.

In addition, I would like to thank Dr. K. Vairavan, Dr. T. Morrissey, Dr. D. Sullivan, Mr. J. Geist, Mr. J. Chang, Mr. W. Hartman, and Mr. R. Olson for many helpful discussions. Gratitude is also due Dr. Vairavan, Mr. Geist, Mr. Chang, Mr. J. Brennan, and Mr. J. Wruck for help in preparing the computer programs used in this thesis.

Finally, I am deeply indebted to the National Science Foundation and to the National Aeronautics and Space Administration for supporting both my studies and my research.

ABSTRACT

A broad look at the problems of convolutional encoding and sequential decoding of digital data for reliable transmission over a noisy channel is presented. Various distance measures for convolutional encoders are carefully defined. A distance measure called the free distance, and introduced by Massey [9], McEliece and Rumsey [10], and Neumann [11], is shown to be a more important parameter for codes used with sequential decoding than the conventional minimum distance.

The notion of equivalent encoders is carefully considered, and a new definition of encoder equivalence which preserves the distance properties of the encoder is given. For those non-systematic encoders that do not have a systematic equivalent which preserves distance properties, a method is presented for converting the encoder to a better systematic encoder. Also, general parity-check matrices and syndrome forming circuits for non-systematic encoders of all rates are presented.

A new lower bound on free distance is given, and McEliece and Rumsey's [10] upper bound is generalized. Also, a new lower bound on definite decoding minimum distance, for both systematic and non-systematic codes, is derived, as well as Gilbert lower bounds on feedback decoding minimum distance for two simply implemented subclasses of convolutional codes. Finally, some new methods for calculating the free distance are discussed.

Construction algorithms are given which produce longer good codes than any previously known. A sequential decoding system was simulated for both the Gaussian and binary symmetric channels. Various codes were tested over different channels for their performance as regards decoding probability of error and decoding speed (number of computations). A simply implemented non-systematic rate 1/2 code was found to be far superior to all other codes tested with the same constraint length.

Finally, a technique introduced by McEliece [10] for constructing good convolutional codes from known block codes was extended, and some convolutional codes discovered by Wyner [34] and Sullivan [35] were discussed.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	i
ABSTRACT	ii
Chapter	
I. INTRODUCTION	1
A. Background.....	1
B. Review of Convolutional Encoding.....	1
C. Review of Algebraic Decoding	12
II. DISTANCE DEFINITIONS FOR CONVOLUTIONAL ENCODERS	24
A. Feedback Decoding Minimum Distance.....	24
B. Definite Decoding Minimum Distance.....	29
C. Free Distance.....	33
D. Reverse Distance.....	42
III. ENCODER EQUIVALENCE AND SYNDROME FORMATION....	43
A. Encoder Equivalence.....	43
B. Syndrome Formation.....	53
IV. BOUNDS ON DISTANCE.....	63
A. Introduction.....	63
B. Bounds on d_{FD}	64
C. Bounds on d_{DD}	65
D. Bounds on d_{FREE}	78
E. A Gilbert Lower Bound for an Easily Instrumented Subclass of $R = 1/2$ Non-Systematic Codes..	102
V. SOME RESULTS ON FREE DISTANCE.....	105
A. Bounding the Length of the Information Sequence Which Produces the Shortest Minimum Free Weight Codeword.....	105
B. Calculating d_{FREE}	109
VI. CONSTRUCTING GOOD CONVOLUTIONAL CODES.....	112
A. The Minimum Weight Construction Algorithms	112
B. More Construction Algorithms for $R = 1/2$ Codes.....	131
C. Performance of Codes with Sequential Decoding.....	142

TABLE OF CONTENTS (Continued)

	Page
VII. DERIVING GOOD CONVOLUTIONAL CODES FROM CYCLIC CODES.....	153
A. McEliece's Codes.....	153
B. Wyner-Sullivan Codes.....	158
C. Forney Canonic Form.....	160
D. Codes Dependent on the Length of Information Sequence Bound.....	163
E. Some Upper Bounds on Free Distance.....	164
VIII. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS FOR FURTHER RESEARCH.....	167
APPENDIX A.....	173
APPENDIX B.....	187
REFERENCES.....	190

I. Introduction

A. Background

Convolutional codes were first introduced by Elias in 1955 [1]. Recently, Viterbi [2] has shown that a lower probability of decoding error is achievable with convolutional codes than with block codes of the same rate and comparable complexity used over a memoryless channel. Another advantage of convolutional codes over block codes is the existence for the former of a general, simply instrumented, and near-optimal decoding algorithm, called sequential decoding, whose evolution began with Wozencraft in 1957 [3]. Sequential decoding is probabilistic in nature, and the same basic sequential decoding algorithm is applicable to a very general class of codes, called tree codes, of which convolutional codes are a subclass. At the present time, sequential decoding of convolutional codes represents the best practical means of employing long codes on the additive white Gaussian noise channel and on many other memoryless channels. Algebraic decoding techniques for convolutional codes, such as Massey's threshold decoding [4], also provide good coding systems for certain channels with memory, and for short codes on memoryless channels.

B. Review of Convolutional Encoding

It is convenient to define a convolutional code after first making precise what is meant by a convolutional encoder. A more general type of encoder is defined first.

Definition 1.1 An (N, K) general convolutional encoder G is an invertible K -input, N -output realizable linear finite-state machine (FSM) which is in the zero state in the infinite past. |

Figure 1.1 shows an (N, K) general convolutional encoder. At each unit of time the encoder transforms the block of K input digits into a block of N output digits. Realizable means that the encoder output cannot depend upon future inputs. Invertibility [5] means that the input can be recovered, perhaps with delay, from the encoder output and is the elementary condition for an encoder to be useful. Without an inverse, decoding could be ambiguous, even in the absence of noise. If G is time-invariant, it is called a fixed convolutional encoder, or simply a convolutional encoder. When the encoder G is time-invariant, definition 1.1 reduces to Forney's [6] definition of a convolutional encoder.

Consider a semi-infinite (or one-sided) sequence a_n, a_{n+1}, \dots of digits over a finite field $GF(q)$, where $-\infty < n < +\infty$. The transform $A(D)$ of this sequence is defined to be the formal power series

$$A(D) = a_n D^n + a_{n+1} D^{n+1} + \dots .$$

If $a_i = 0$ for all $i < 0$, then the sequence is called causal, and if the sequence has only a finite number of non-zero digits, it is called finite. A polynomial in the indeterminate D over a finite field $GF(q)$ is thus seen to be identified as the transform of a causal, finite sequence. The degree of a polynomial is the highest power of D in the



Fig. 1.1. An (N, K) general convolutional encoder.

polynomial, and the delay of a polynomial is the highest power of D which it contains as a factor. By convention, the polynomial 0 has degree $-\infty$ and delay $+\infty$.

A rational function is a ratio of two polynomials $\frac{P(D)}{Q(D)}$, where $Q(D) \neq 0$ and the division is assumed to begin with the lowest degree terms of each polynomial. A sequence is called rational if its transform is a rational function. The rational sequence with transform $\frac{P(D)}{Q(D)}$ is causal if and only if $n' \leq n$, where n' is the delay of $Q(D)$ and n is the delay of $P(D)$. Causal rational sequences are called realizable since they can be produced by an autonomous linear finite state machine. A realizable function is the transform of a causal rational sequence.

The K input sequences to any encoder will be restricted to be rational sequences. The set of K input sequences will be represented by the semi-infinite row vector $\underline{x} = [x_n, x_{n+1}, x_{n+2}, \dots] = [x_n^{(1)} \dots x_n^{(K)}, x_{n+1}^{(1)} \dots x_{n+1}^{(K)}, x_{n+2}^{(1)} \dots x_{n+2}^{(K)}, \dots]$, where n is the least integer such that $x_n \neq 0$, $x_i = [x_i^{(1)} \dots x_i^{(K)}]$ is the row vector or K -tuple of input digits at time unit i , and $x_i^{(j)}$ is the input digit at time unit i in the j^{th} input sequence, $n \leq i < \infty$, $i \leq j \leq K$. \underline{x} is called the information sequence since it represents the data to be encoded. The transform of the information sequence will be written as the K -tuple $\underline{x}(D) = [x^{(1)}(D), x^{(2)}(D), \dots, x^{(K)}(D)]$, where $x^{(j)}(D) = x_n^{(j)} D^n + x_{n+1}^{(j)} D^{n+1} + x_{n+2}^{(j)} D^{n+2} + \dots$, $1 \leq j \leq K$, is the transform of the j^{th} input sequence.

Since all encoders are realizable and the input sequences are rational, the N output sequences of any encoder are also rational and may be represented by the semi-infinite row vector $\underline{y} = [y_n, y_{n+1}, y_{n+2}, \dots] = [y_n^{(1)} \dots y_n^{(N)}]$, where $y_i = [y_i^{(1)} \dots y_i^{(N)}]$ is the row vector or N-tuple of output digits at time unit i, and $y_i^{(j)}$ is the output digit at time unit i in the j^{th} output sequence, $n \leq i < \infty$, $1 \leq j \leq N$. \underline{y} is called the transmitted sequence or codeword since it represents the data which is actually to be transmitted over the channel.

The transform of the transmitted sequence will be written as the N-tuple $\underline{y}(D) = [y^{(1)}(D), y^{(2)}(D), \dots, y^{(N)}(D)]$, where $y^{(j)}(D) = y_n^{(j)} D^n + y_{n+1}^{(j)} D^{n+1} + y_{n+2}^{(j)} D^{n+2} + \dots$, $1 \leq j \leq N$, is the transform of the j^{th} output sequence.

The encoding equations for a general convolutional encoder over the set of all rational input sequences can be written as

$$\underline{y}_i = \sum_{\ell=0}^{\infty} G_{\ell}^{(i)} \underline{x}_{i-\ell}, \quad (1)$$

where $G_{\ell}^{(i)}$ is a $K \times N$ matrix of elements from $GF(q)$, $0 \leq \ell < \infty$, $n \leq i < \infty$, and all operations are performed over $GF(q)$.

In the remainder of this thesis only causal rational sequences will be allowed as inputs, except when it is explicitly stated that all rational sequences are to be considered as inputs. In this case $n \geq 0$ and the information and transmitted sequences will be assumed to begin at time unit 0, i.e., $\underline{x} = [x_0, x_1, x_2, \dots] = [x_0^{(1)} \dots x_0^{(K)} \ x_1^{(1)} \dots$

$x_1^{(K)} \dots]$ and $\underline{x} = [x_0, x_1, x_2, \dots] = [y_0^{(1)}, \dots, y_0^{(N)}, y_1^{(1)}, \dots, y_1^{(N)}, y_2^{(1)}, \dots, y_2^{(N)}, \dots]$. Hence the encoding equations for a general convolutional encoder over the set of all causal rational input sequences can be written as

$$\underline{y} = \underline{x} \underline{G}, \quad (2)$$

where

$$\underline{G} = \begin{bmatrix} G_0^{(0)} & G_1^{(0)} & G_2^{(0)} & \dots \\ 0 & G_0^{(1)} & G_1^{(1)} & G_2^{(1)} & \dots \\ 0 & 0 & G_0^{(2)} & G_1^{(2)} & G_2^{(2)} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (3)$$

is called the generator matrix and

$$G_\ell^{(u)} = \begin{bmatrix} g_{\ell 1}^{(1)}(u) & g_{\ell 1}^{(2)}(u) & \dots & g_{\ell 1}^{(N)}(u) \\ g_{\ell 2}^{(1)}(u) & g_{\ell 2}^{(2)}(u) & \dots & g_{\ell 2}^{(N)}(u) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ g_{\ell K}^{(1)}(u) & g_{\ell K}^{(2)}(u) & \dots & g_{\ell K}^{(N)}(u) \end{bmatrix} \quad (4)$$

is a $K \times N$ matrix of elements from $GF(q)$, $0 \leq u, \ell < \infty$.

Definition 1.1 requires that each sequence $g_{0i}^{(j)}(u)$, $g_{1i}^{(j)}(u)$, $g_{2i}^{(j)}(u), \dots$, $1 \leq i \leq K$, $1 \leq j \leq N$, $0 \leq u < \infty$, be realizable.

These sequences are called generator sequences.

The generator matrix for a fixed convolutional encoder is written as

$$\underline{G} = \begin{bmatrix} G_0 & G_1 & G_2 & \dots \\ 0 & G_0 & G_1 & G_2 & \dots \\ 0 & 0 & G_0 & G_1 & G_2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad (5)$$

where

$$\underline{g}_\ell = \begin{bmatrix} g_{\ell 1}^{(1)} & g_{\ell 1}^{(2)} & \dots & g_{\ell 1}^{(N)} \\ g_{\ell 2}^{(1)} & g_{\ell 2}^{(2)} & \dots & g_{\ell 2}^{(N)} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ g_{\ell K}^{(1)} & g_{\ell K}^{(2)} & \dots & g_{\ell K}^{(N)} \end{bmatrix}, \quad 0 \leq \ell < \infty, \quad (6)$$

and each generator sequence $g_{0i}^{(j)}, g_{1i}^{(j)}, g_{2i}^{(j)}, \dots$

$1 \leq i \leq K, 1 \leq j \leq N$, is realizable. It is often convenient to represent a fixed convolutional encoder by a $K \times N$ matrix $\underline{G}(D)$, where

$$\underline{G}(D) = \begin{bmatrix} G_1^{(1)}(D) & G_1^{(2)}(D) & \dots & G_1^{(N)}(D) \\ G_2^{(1)}(D) & G_2^{(2)}(D) & \dots & G_2^{(N)}(D) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ G_K^{(1)}(D) & G_K^{(2)}(D) & \dots & G_K^{(N)}(D) \end{bmatrix} \quad (7)$$

and

$$G_i^{(j)}(D) = g_{0i}^{(j)} + g_{1i}^{(j)} D + g_{2i}^{(j)} D^2 + \dots \quad (8)$$

is the transform of the generator sequence $g_{01}^{(j)}, g_{1i}^{(j)}, g_{2i}^{(j)}, \dots, 1 \leq i \leq K, 1 \leq j \leq N$. Each $G_i^{(j)}(D)$ is called a generator function. Then the encoding equations can be written in D-operator form as

$$\underline{y}(D) = \underline{x}(D) \underline{G}(D), \quad (9)$$

where all operations are performed over $GF(q)$.

The following definitions are due to Forney [6] and apply to both time-varying and fixed convolutional codes.

Definition 1.2 The convolutional code C generated by a convolutional encoder G is the set of all output sequences \underline{y} of G produced by the set of all rational input sequences \underline{x} to G . |

Definition 1.3 Two encoders are said to be equivalent if they generate the same convolutional code. |

The rate R of a general convolutional encoder is defined to be $\frac{K}{N}$. The time- u memory order $m(u)$ of a general convolutional encoder is defined as

$$m(u) = \begin{cases} \max_{0 \leq i < \infty} \{ i \mid G_i(u) \neq 0 \} & \\ \infty & \text{if no such } i \text{ exists.} \end{cases} \quad (10)$$

The memory order m of the encoder is then defined as

$$m = \begin{cases} \max_{0 \leq u < \infty} \{ m(u) \} & \text{if } m(u) \text{ is finite for all } u \\ \infty & \text{otherwise.} \end{cases} \quad (11)$$

For fixed convolutional encoders, the memory order m of the encoder is defined as

$$m = \begin{cases} \max_{0 \leq i < \infty} \{ i \mid G_i \neq 0 \} & \\ \infty & \text{if no such } i \text{ exists.} \end{cases} \quad (12)$$

All encoders with $m = 0$ are block encoders. Therefore convolutional codes include block codes as a special case.

Let u_{\max} be the least value of u such that $m(u)$ is a maximum if $m(u)$ is finite for all $u \geq 0$. Otherwise let u_{\max} be the least value of u such that $m(u)$ is infinite. Then the number of encoded digits in those encoded blocks inclusive between the first and last encoded block which depend on the time- u_{\max} block of input digits is defined to be the encoding constraint length n_A . For both general and fixed convolutional encoders of memory order m ,

$$n_A = N(m + 1). \quad (13)$$

For time-varying encoders of memory order m , the generator matrix can be rewritten as

$$\underline{G} = \begin{bmatrix} G_0(0) & G_1(0) & G_2(0) & \dots & G_m(0) & 0 & 0 & \dots \\ 0 & G_0(1) & G_1(1) & \dots & G_{m-1}(1) & G_m(1) & 0 & \dots \\ 0 & 0 & G_0(2) & \dots & G_{m-2}(2) & G_{m-1}(2) & G_m(2) & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (14)$$

and for fixed encoders of memory order m , the generator matrix becomes

$$\underline{G} = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_m & 0 & 0 & \dots \\ 0 & G_0 & G_1 & \dots & G_{m-1} & G_m & 0 & \dots \\ 0 & 0 & G_0 & \dots & G_{m-2} & G_{m-1} & G_m & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (15)$$

If

$$G_0(u) = [I_K : Q_0(u)], \quad 0 \leq u < \infty, \quad (16)$$

where I_K is the $K \times K$ identity matrix and $\underline{Q}_0(u)$ is a $K \times (N - K)$ matrix of elements from $GF(q)$, and

$$\underline{G}_i(u) = [0_K : \underline{Q}_i(u)] , \quad 0 \leq u < \infty , \quad 1 \leq i < \infty , \quad (17)$$

where 0_K is the $K \times K$ all-zero matrix and $\underline{Q}_i(u)$ is a $K \times (N - K)$ matrix of elements from $GF(q)$, then the encoder is said to be in canonic systematic form. An encoder is said to be in systematic form if the output terminals can be re-numbered so that the resultant encoder is in canonic systematic form. For systematic encoders some set of K output sequences are reproductions of the K input sequences, and for canonic systematic encoders, the transmitted sequence y can be written as

$$y = [x_0, p_0, x_1, p_1, \dots] \quad (18)$$

$$= [x_0^{(1)} \dots x_0^{(K)} p_0^{(K+1)} \dots p_0^{(N)} x_1^{(1)} \dots \\ x_1^{(K)} p_1^{(K+1)} \dots p_1^{(N)} \dots] , \quad (19)$$

where p_i is an $(N - K)$ -dimensional row vector called a parity vector, and $p_i^{(j)}$ is called a parity digit, $0 \leq i < \infty$, $K + 1 \leq j \leq N$.

A fixed code is in canonic systematic form if

$$\underline{G}_0 = [I_K : \underline{Q}_0] \quad (20)$$

and

$$\underline{G}_i = [0_K : \underline{Q}_i] , \quad 1 \leq i < \infty . \quad (21)$$

This is equivalent to requiring that

$$\underline{G}(D) = [I_K : \underline{Q}(D)] , \quad (22)$$

where

$$\underline{Q}(D) = Q_0 + Q_1 D + Q_2 D^2 + \dots . \quad (23)$$

In this case the transform $\underline{y}(D)$ of the transmitted sequence can be written as

$$\underline{y}(D) = [x^{(1)}(D), \dots, x^{(K)}(D), p^{(K+1)}(D), \dots, p^{(N)}(D)], \quad (24)$$

where

$$p^{(j)}(D) = p_0^{(j)} + p_1^{(j)}D + p_2^{(j)}D^2 + \dots \quad (25)$$

is the transform of the j^{th} parity sequence $p_0^{(j)}, p_1^{(j)}, p_2^{(j)}, \dots, K+1 \leq j \leq N$.

For a time-varying convolutional encoder, if

$$\underline{G}_i(u) = \underline{G}_i(u + T), \quad 0 \leq i < \infty, \quad 0 \leq u < \infty, \quad (26)$$

the encoder is called periodic with period T . Clearly a fixed encoder is a periodic encoder with period $T = 1$.

The most familiar convolutional encoders are the $R = \frac{1}{N}$ fixed encoders. In this case the first row of the generator matrix \underline{G} is called the generator, and is labeled \underline{g} . Figure 1.2 shows an example of what Forney calls the obvious realization [6] of a binary $R = 1/2$ canonic systematic fixed convolutional encoder with $\underline{G}(D) = \left[1 \quad \frac{1 + D + D^2}{1 + D} \right]$. Note that the encoder is realized with two memory elements even though the memory order of the encoder is infinite. When $\underline{G}(D)$ contains only polynomial elements the number of memory elements required in the obvious realization is equal to the memory order of the encoder. However, a rational function $\frac{P(D)}{Q(D)}$ requires only $\max \{ \text{degree}[P(D)], \text{degree}[Q(D)] \}$ memory elements in its realization. The feedback used to realize a

rational function accounts for the infinite memory order of the encoder.

It is usually desirable to eliminate feedback in the encoder so that an encoding error caused by an equipment malfunction will not propagate (cause other errors) throughout the transmitted sequence. Feedback can be eliminated by multiplying each row of $\underline{G}(D)$ by the least common multiple of the denominators of its generator functions resulting in a matrix $\underline{G}'(D)$ with only polynomial elements. The resulting encoder is equivalent to the original encoder. For example, the encoder given above has $\underline{G}'(D) = \begin{bmatrix} 1 + D & 1 + D + D^2 \end{bmatrix}$. An input $\underline{x}(D) = \begin{bmatrix} x^{(1)}(D) \\ x(D) \end{bmatrix} = x(D)$ to $\underline{G}(D)$ produces the same codeword as an input $\frac{x(D)}{1+D}$ to $\underline{G}'(D)$. The obvious realization of $\underline{G}'(D)$ is shown in Figure 1.3. Note that the polynomial encoder has memory order two but is not canonic systematic.

C. Review of Algebraic Decoding

Let

$$\underline{H} = \begin{bmatrix} \underline{H}_0(0) & 0 & 0 \\ \underline{H}_1(0) & \underline{H}_0(1) & 0 \\ \underline{H}_2(0) & \underline{H}_1(1) & \underline{H}_0(2) \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (27)$$

be a semi-infinite matrix and

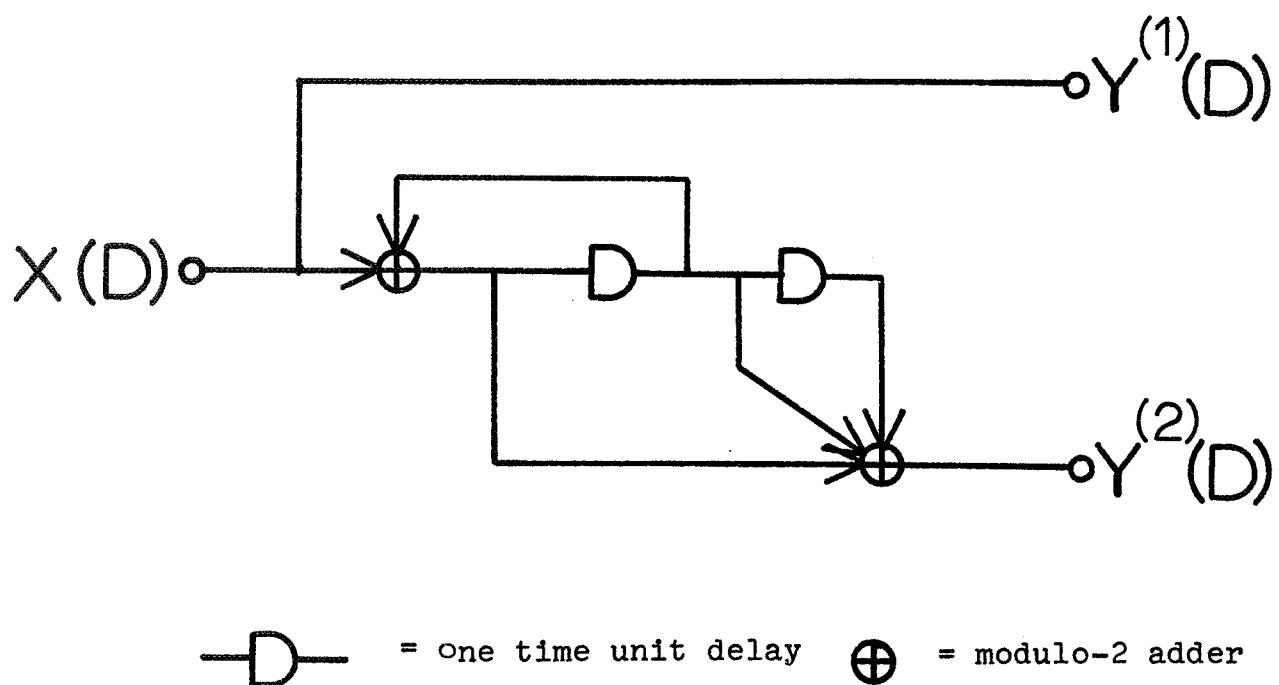


Fig. 1.2. Obvious realization of a rational function encoder.

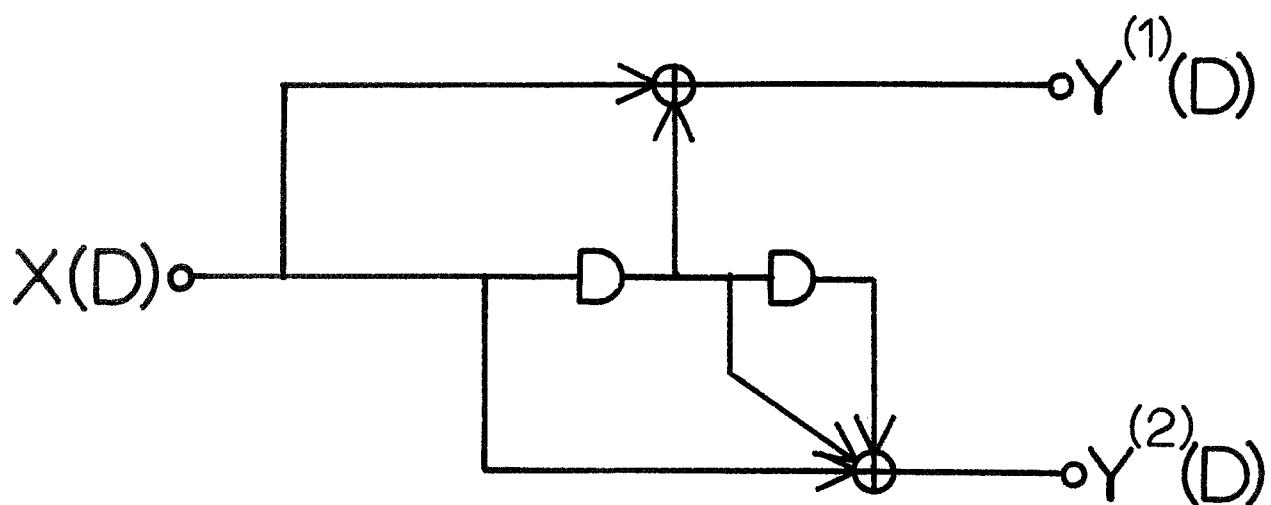


Fig. 1.3. Obvious realization of a polynomial encoder.

$$\underline{H}_{\ell}(u) = \begin{bmatrix} h_{\ell,1}^{(1)}(u) & h_{\ell,1}^{(2)}(u) \dots h_{\ell,1}^{(N)}(u) \\ h_{\ell,2}^{(1)}(u) & h_{\ell,2}^{(2)}(u) \dots h_{\ell,2}^{(N)}(u) \\ \vdots & \vdots \\ \vdots & \vdots \\ h_{\ell,N-K}^{(1)}(u) & h_{\ell,N-K}^{(2)}(u) \dots h_{\ell,N-K}^{(N)}(u) \end{bmatrix} \quad (28)$$

be an $(N - K) \times N$ matrix of elements from $GF(q)$, $0 \leq u, \ell < \infty$, such that

$$\text{rank } \left[\underline{H}_0(u) \right] = N - K, \quad 0 \leq u < \infty, \quad (29)$$

$$\underline{G} \underline{H}^T = \underline{0}, \quad (30)$$

where \underline{H}^T is the transpose of \underline{H} , $\underline{0}$ is a semi-infinite all-zero matrix, and each sequence $h_{0i}^{(j)}(u)$, $h_{1i}^{(j)}(u)$, $h_{2i}^{(j)}(u)$, ..., $1 \leq i \leq N - K$, $1 \leq j \leq N$, $0 \leq u < \infty$, is realizable.

These sequences are called parity-check sequences and \underline{H} is called a parity-check matrix.

The parity-check matrix for fixed convolutional encoders is written as

$$\underline{H} = \begin{bmatrix} \underline{H}_0 & \underline{0} & \underline{0} \\ \underline{H}_1 & \underline{H}_0 & \underline{0} \\ \underline{H}_2 & \underline{H}_1 & \underline{H}_0 \\ \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad (31)$$

where

$$\underline{H}_{\ell} = \begin{bmatrix} h_{\ell,1}^{(1)} & h_{\ell,1}^{(2)} & \dots & h_{\ell,1}^{(N)} \\ h_{\ell,2}^{(1)} & h_{\ell,2}^{(2)} & \dots & h_{\ell,2}^{(N)} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ h_{\ell,N-K}^{(1)} & h_{\ell,N-K}^{(2)} & \dots & h_{\ell,N-K}^{(N)} \end{bmatrix}, 0 \leq \ell < \infty, \quad (32)$$

$$\text{rank } [\underline{H}_0] = N - K, \quad (33)$$

and each parity-check sequence $h_{0i}^{(j)}, h_{1i}^{(j)}, h_{2i}^{(j)}, \dots, 1 \leq i \leq N - K, 1 \leq j \leq N$, is realizable. It is often convenient to introduce an $(N - K) \times N$ matrix $\underline{H}(D)$, where

$$\underline{H}(D) = \begin{bmatrix} H_1^{(1)}(D) & H_1^{(2)}(D) & \dots & H_1^{(N)}(D) \\ H_2^{(1)}(D) & H_2^{(2)}(D) & \dots & H_2^{(N)}(D) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ H_{N-K}^{(1)}(D) & H_{N-K}^{(2)}(D) & \dots & H_{N-K}^{(N)}(D) \end{bmatrix} \quad (34)$$

and

$$H_i^{(j)}(D) = h_{0i}^{(j)} + h_{1i}^{(j)}D + h_{2i}^{(j)}D^2 + \dots \quad (35)$$

is the transform of the parity-check sequence $h_{0i}^{(j)}, h_{1i}^{(j)}, h_{2i}^{(j)}, \dots, 1 \leq i \leq N - K, 1 \leq j \leq N$. Each $H_i^{(j)}(D)$ is called a parity-check function. Then equations (30), the parity-check equations, can be written in D-operator form as

$$\underline{G}(D) \underline{H}^T(D) = \underline{0}, \quad (36)$$

where $\underline{0}$ is the $K \times (N-K)$ all-zero matrix.

For any transmitted sequence \underline{x}

$$\underline{x} \underline{H}^T = \underline{x} \underline{G} \underline{H}^T = \underline{x} \underline{0} = \underline{0}, \quad (37)$$

where $\underline{0}$ is a semi-infinite all-zero vector. For fixed encoders equation (37) can be written in D-operator form as

$$\underline{y}(D) \underline{H}^T(D) = \underline{x}(D) \underline{G}(D) \underline{H}^T(D) = \underline{x}(D) \underline{0} = \underline{0}, \quad (38)$$

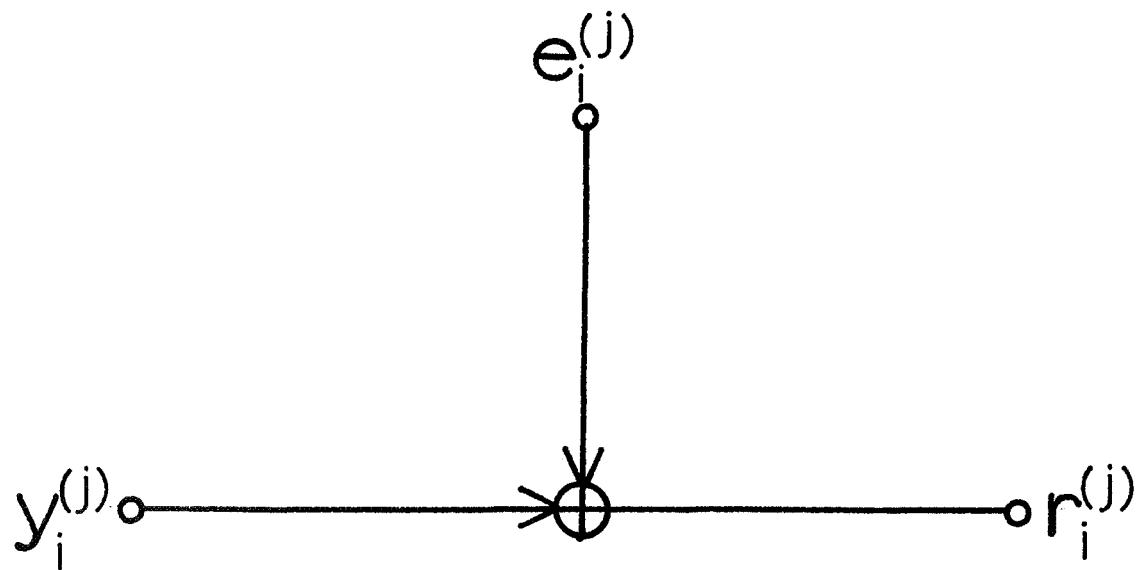
where in this case $\underline{0}$ is the $(N - K)$ -dimensional all-zero vector. Therefore every transmitted sequence is in the null space of \underline{H}^T . However if $\text{rank}[\underline{H}_0] < N - K$, the encoder has no zero-delay inverse [5], and the null space of \underline{H}^T contains other sequences besides the set of codewords.

Note that

$$\underline{H}' = \begin{bmatrix} \underline{H}_0 & \underline{H}_1 & \underline{H}_2 & \underline{H}_3 & \underline{H}_4 & \underline{H}_5 & \dots \\ \underline{0} & \underline{H}_0 & \underline{H}_1 & \underline{H}_2 & \underline{H}_3 & \underline{H}_4 & \dots \\ \underline{0} & \underline{0} & \underline{H}_0 & \underline{H}_1 & \underline{H}_2 & \underline{H}_3 & \dots \end{bmatrix} \quad (39)$$

is a generator matrix for an $R = \frac{N - K}{N}$ encoder. This encoder is said to be dual to the encoder whose generator matrix is \underline{G} . For fixed encoders, $\underline{H}(D)$ is the matrix of generator functions for the dual encoder.

Figure 1.4 shows a simplified picture of a channel with additive noise. Let the noise be represented by an error sequence $\underline{e} = [e_0, e_1, e_2, \dots] = [e_0^{(1)}, \dots, e_0^{(N)}, e_1^{(1)}, \dots, e_1^{(N)}, e_2^{(1)}, \dots, e_2^{(N)}, \dots]$, where $e_i^{(j)} \in GF(q)$ for all i and j . The transform of the error sequence is $\underline{e}(D) = [e^{(1)}(D), \dots, e^{(N)}(D)]$, where $e^{(j)}(D) = e_0^{(j)} + e_1^{(j)}D + e_2^{(j)}D^2 + \dots$, $1 \leq j \leq N$. Similarly the received sequence $\underline{r} = [r_0, r_1, r_2, \dots] = [r_0^{(1)}, \dots, r_0^{(N)}, r_1^{(1)}, \dots, r_1^{(N)}, r_2^{(1)}, \dots, r_2^{(N)}, \dots]$ and the transform of the received sequence is $\underline{r}(D) = [r^{(1)}(D), \dots, r^{(N)}(D)]$. Since the noise is additive, the received sequence is the modulo-q sum of the transmitted sequence and the error sequence, i.e.,



\oplus = modulo-q adder

Fig. 1.4. Additive noise channel.

$$\underline{r} = \underline{y} + \underline{e} . \quad (40)$$

In terms of transforms, equation (40) is

$$\underline{r}(D) = \underline{y}(D) + \underline{e}(D) . \quad (41)$$

Clearly, for all i and j ,

$$r_i^{(j)} \neq y_i^{(j)} \text{ if and only if } e_i^{(j)} \neq 0. \quad (42)$$

Thus $e_i^{(j)} \neq 0$ corresponds to an error in transmission, whereas $e_i^{(j)} = 0$ corresponds to a correctly transmitted digit.

A common form of algebraic decoding computes the syndrome sequence $\underline{s} = [s_0, s_1, \dots] = [s_0^{(1)}, \dots, s_0^{(N-K)}, s_1^{(1)}, \dots, s_1^{(N-K)}, \dots]$ at the receiver from the parity-check matrix as follows:

$$\underline{s} = \underline{r} \underline{H}^T . \quad (43)$$

From equations (37) and (40) it can easily be deduced that

$$\underline{s} = \underline{e} \underline{H}^T . \quad (44)$$

In terms of transforms $\underline{s}(D) = [s^{(1)}(D), \dots, s^{(N-K)}(D)]$, where $s^{(j)}(D) = s_0^{(j)} + s_1^{(j)} D + s_2^{(j)} D^2 + \dots, 1 \leq j \leq N - K$, and for fixed encoders equation (44) can be written as

$$\underline{s}(D) = \underline{e}(D) \underline{H}^T(D) . \quad (45)$$

Therefore the syndrome depends only on the errors and not on the particular codeword transmitted. Any decoding function which estimates the error sequence from the syndrome is called a syndrome decoder. Equation (44) and (45) are referred to as the syndrome equations.

It is often necessary to write the encoding equations over only one constraint length of transmitted digits. If

$\underline{z} = [z_0, z_1, z_2, \dots]$, then let $[\underline{z}]_h$ denote the truncated vector $[z_0, z_1, \dots, z_h]$. Then the encoding equations for the first constraint length are

$$[\underline{y}]_m = [\underline{x}]_m \begin{bmatrix} G_0(0) & G_1(0) & \dots & G_m(0) \\ 0 & G_0(1) & \dots & G_{m-1}(1) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & G_0(m) \end{bmatrix} \quad (46)$$

$$= [\underline{x}]_m [G]_m \quad (47)$$

$$= [\underline{x} \underline{G}]_m, \quad (48)$$

where $[G]_m$ indicates that G has been truncated after $(m + 1)N$ columns.

In syndrome decoding it is usually assumed that the error digits at time unit u , \underline{e}_u , are estimated by looking at the syndrome digits $s_u, s_{u+1}, \dots, s_{u+m}$ from time unit u through time unit $u + \bar{m}$, $0 \leq u < \infty$. \bar{m} is called the decoding memory, and $\bar{n}_A = N(\bar{m} + 1)$ is called the decoding constraint length. Therefore the first constraint length of syndrome equations can be written as

$$[\underline{s}]_{\bar{m}} = [\underline{e}]_{\bar{m}} \begin{bmatrix} H_0^T(0) & H_1^T(0) & \dots & H_{\bar{m}}^T(0) \\ 0 & H_0^T(1) & \dots & H_{\bar{m}-1}^T(1) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & H_0^T(\bar{m}) \end{bmatrix} \quad (49)$$

$$= [\underline{e}] \quad \bar{\underline{m}} \quad \left[\underline{H}^T \right] \quad \bar{\underline{m}} \quad (50)$$

$$= \left[\underline{e} \quad \underline{H}^T \right] \quad \bar{\underline{m}}, \quad (51)$$

where $\left[\underline{H}^T \right]_{\bar{\underline{m}}}$ indicates that \underline{H}^T has been truncated after $(\bar{\underline{m}} + 1)(N - K)$ columns. $\bar{\underline{m}}$ is often chosen as

$$\bar{\underline{m}} = \begin{cases} \max_{0 \leq u < \infty} \max_{0 \leq i < \infty} \{ i | \underline{H}_i(u) \neq \underline{0} \} \\ \infty \quad \text{if no such } i \text{ exists.} \end{cases} \quad (52)$$

For fixed encoders equation (52) reduces to

$$\bar{\underline{m}} = \begin{cases} \max_{0 \leq i < \infty} \{ i | \underline{H}_i \neq \underline{0} \} \\ \infty \quad \text{if no such } i \text{ exists.} \end{cases} \quad (53)$$

Choosing $\bar{\underline{m}}$ in this way corresponds to defining the decoding constraint length \bar{n}_A as the number of error digits inclusive between the first and last blocks of error digits which affect the syndrome at time $u_{\max} + \bar{\underline{m}}$, where u_{\max} is the least value of u which maximizes (52).

For canonic systematic encoders, the generator matrix can be written as

$$\underline{G} = \begin{bmatrix} I_K : \underline{Q}_0(0) : 0_K : \underline{Q}_1(0) : 0_K : \underline{Q}_2(0) : \dots \\ 0 \quad I_K : \underline{Q}_0(1) : 0_K : \underline{Q}_1(1) : \dots \\ 0 \quad 0 \quad I_K : \underline{Q}_0(2) : \dots \\ \ddots \end{bmatrix} \quad (54)$$

Clearly, it follows that

$$\begin{bmatrix} I_K : \underline{\Omega}_0(0) : 0_K : \underline{\Omega}_1(0) : 0_K : \underline{\Omega}_2(0) : \dots \\ 0 & I_K : \underline{\Omega}_0(1) : 0_K : \underline{\Omega}_1(1) : \dots \\ 0 & 0 & I_K : \underline{\Omega}_0(2) : \dots \\ \ddots & \ddots & \ddots \end{bmatrix} \quad \begin{bmatrix} -\underline{\Omega}_0(0) & -\underline{\Omega}_1(0) & -\underline{\Omega}_2(0) & \dots \\ I_{N-K} & 0_{N-K} & 0_{N-K} & \dots \\ -\underline{\Omega}_0(1) & -\underline{\Omega}_1(1) & \dots & = 0 \\ 0 & I_{N-K} & 0_{N-K} & \dots \\ -\underline{\Omega}_0(2) & \dots & \ddots & \ddots \\ 0 & 0 & I_{N-K} & \dots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix} \quad (55)$$

where I_{N-K} is the $(N - K) \times (N - K)$ identity matrix and

0_{N-K} is the $(N - K) \times (N - K)$ all-zero matrix. Since

$$\text{rank} \left[I_{N-K} : -\underline{\Omega}_0^T(u) \right] = N - K, \quad 0 \leq u < \infty, \quad (56)$$

$$H = \begin{bmatrix} -\underline{\Omega}_0^T(0) : I_{N-K} & 0 & 0 \\ -\underline{\Omega}_1^T(0) : 0_{N-K} : -\underline{\Omega}_0^T(1) : I_{N-K} & 0 \\ -\underline{\Omega}_2^T(0) : 0_{N-K} : -\underline{\Omega}_1^T(1) : 0_{N-K} : -\underline{\Omega}_0^T(2) : I_{N-K} & \dots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (57)$$

is a valid parity-check matrix for the encoder with generator matrix \underline{G} . For fixed canonic systematic encoders

$$\underline{G} = \begin{bmatrix} I_K : \underline{\Omega}_0 : 0_K : \underline{\Omega}_1 : 0_K : \underline{\Omega}_2 : \dots \\ 0 : I_K : \underline{\Omega}_0 : 0_K : \underline{\Omega}_1 : \dots \\ 0 & 0 & I_K : \underline{\Omega}_0 : \dots \\ \ddots & \ddots & \ddots \end{bmatrix} \quad (58)$$

and

$$\underline{H} = \begin{bmatrix} -\underline{\Omega}_0^T : I_{N-K} & 0 & 0 \\ -\underline{\Omega}_1^T : 0_{N-K} : -\underline{\Omega}_0^T : I_{N-K} & 0 \\ \vdots & \vdots & \vdots \\ -\underline{\Omega}_2^T : 0_{N-K} : -\underline{\Omega}_1^T : 0_{N-K} : -\underline{\Omega}_0^T : I_{N-K} & \ddots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (59)$$

is a valid parity-check matrix. Therefore if the decoding memory \bar{m} is chosen according to equation (52) for periodic encoders or equation (53) for fixed encoders, it can be seen that $\bar{m} = m$ for systematic encoders and hence $\bar{n}_A = n_A$. \underline{H} will be related to \underline{G} for non-systematic encoders in Chapter III.

The two commonly used modes of algebraic decoding have been termed feedback decoding and definite decoding by Robinson [7]. Although usually used with syndrome decoding, these two decoding modes can be used with any algebraic decoding technique.

If $\underline{z} = [z_0, z_1, z_2, \dots]$, then let $[\underline{z}]_{h, h+\ell}$ denote the doubly truncated vector $[z_h, z_{h+1}, \dots, z_{h+\ell}]$. Then the syndrome equations from time unit u through time unit $u + \bar{m}$ can be written as

$$[\underline{s}]_{u, u+\bar{m}} = [\underline{e}]_{u-\bar{m}, u+\bar{m}} \begin{bmatrix} \underline{H}_{\bar{m}}^T(u-\bar{m}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots \\ \underline{H}_0^T(u) & \dots & \underline{H}_{\bar{m}}^T(u) \\ \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots \\ 0 & \dots & \underline{H}_0^T(u+\bar{m}) \end{bmatrix} \quad (60)$$

In feedback decoding, at time unit $u + \bar{m}$ all blocks of error digits $\underline{e}_0, \underline{e}_1, \dots, \underline{e}_{u-1}$ up to time unit u are assumed to have been decoded correctly and are "fed back" and subtracted out of each syndrome equation in which they appear. Therefore equations (60) become

$$[\underline{s}]_{u, u+\bar{m}} = [\underline{e}]_{u, u+\bar{m}} \begin{bmatrix} \underline{H}_0^T(u) & \dots & \underline{H}_{\bar{m}}^T(u) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ 0 & \dots & \underline{H}_0^T(u+\bar{m}) \end{bmatrix} \quad (61)$$

and \underline{e}_u is estimated from equations (61). In feedback decoding, the decoding memory \bar{m} is always chosen to be m and the feedback decoding constraint length $n_{FD} = N(M + 1) = n_A$. In definite decoding, the error digits up to time unit u are not "fed back" and \underline{e}_u is estimated directly from equations (60). The definite decoding constraint length n_{DD} is sometimes chosen to be the number of error digits which can affect equations (60) for $u \geq \bar{m}$, i.e., $n_{DD} = N(2\bar{m} + 1)$, but it is often useful to choose it otherwise, as will be seen in later chapters. Figures 1.5 and 1.6 show a syndrome decoder operating in the feedback and definite decoding modes, respectively, for the binary fixed $R = 1/2$ encoder with $\underline{G}(D) = [1 \ 1 + D]$.

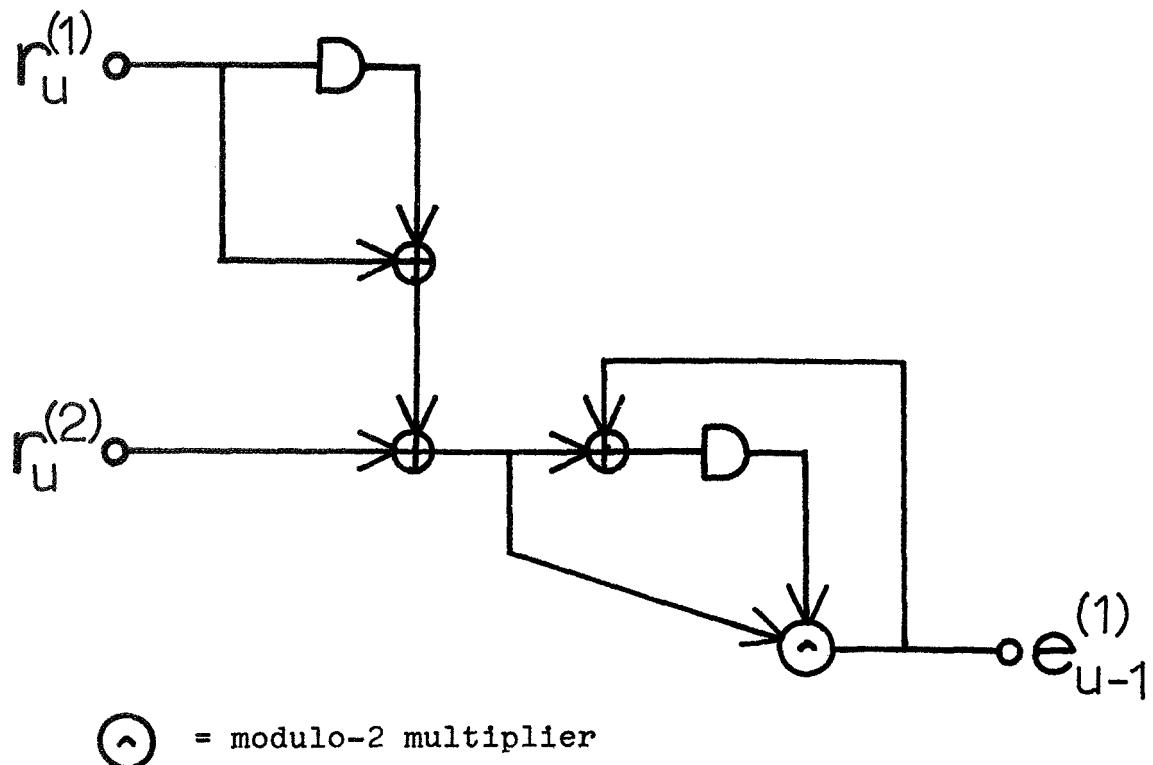


Fig. 1.5 A syndrome feedback decoder.

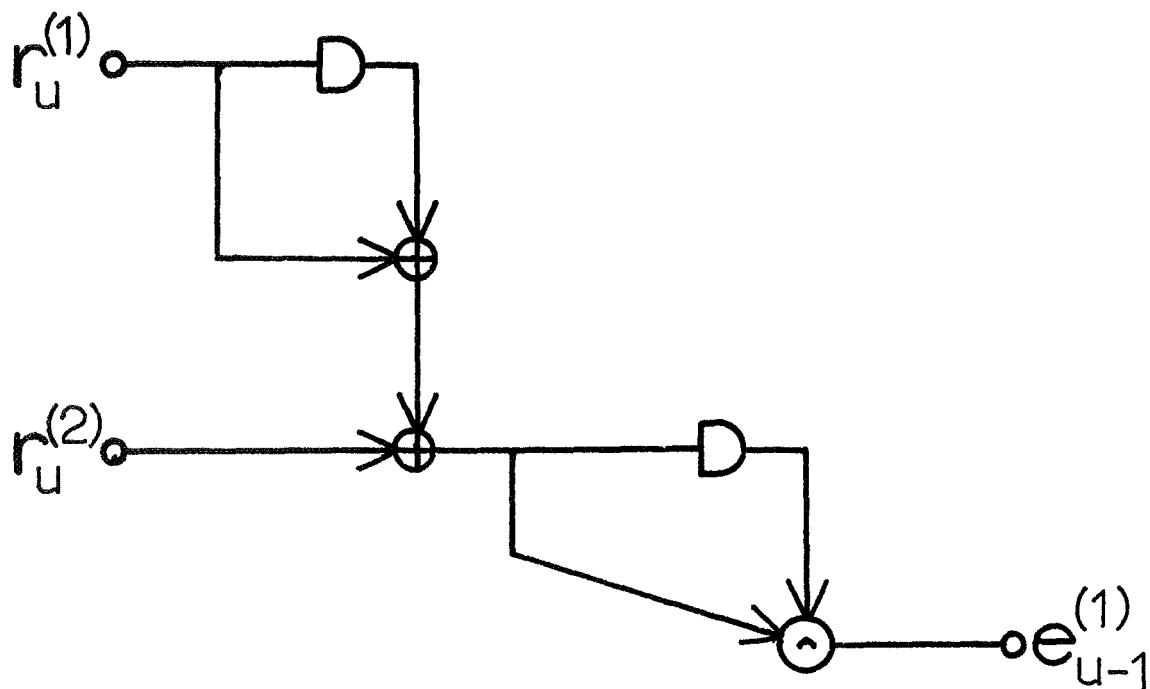


Fig. 1.6 A syndrome definite decoder.

II. Distance Definitions for Convolutional Encoders

In this chapter a number of different distance measures will be defined for convolutional encoders. Distances between codewords in a convolutional code are closely connected with the probability of decoding error. For instance, it is well known [3] that for an encoder with feedback decoding minimum distance d_{FD} (formerly called d_{min}), $\left\lceil \frac{d_{FD} - 1}{2} \right\rceil$ errors within a feedback decoding constraint length are guaranteed correctable by an algebraic decoder operating in the feedback decoding mode, where $[I]$ is the largest integer less than or equal to I . Hence it is usually desirable to design an encoder with good distance properties. For convenience, in the remainder of this thesis only binary codes will be considered, although many of the results apply to codes defined over larger alphabets.

A. Feedback Decoding Minimum Distance

The standard definition of distance for convolutional encoders, feedback decoding minimum distance, d_{FD} , will now be generalized to time-varying encoders.

Definition 2.1

$$d_{FD} = \min_{0 \leq u < \infty} \min_{\substack{x_u \neq x'_u \\ [x]_{u-1} = [x']_{u-1}}} d_H([xG]_{u,u+m}, [x'G]_{u+m}) =$$

$$[x'G]_{u,u+m} = \min_{0 \leq u < \infty} \min_{\substack{x_u \neq x'_u \\ [x]_{u-1} = [x']_{u-1}}} d_H([xG]_{u+m}, [x'G]_{u+m}) =$$

$$\min_{0 \leq u < \infty} \min_{\substack{x_u \neq x'_u \\ [x]_{u-1} = [x']_{u-1}}} d_H([xG]_{u+m}, [x'G]_{u+m}) = \min_{0 \leq u < \infty} \min_{\substack{x_u \neq x'_u \\ [x]_{u-1} = [x']_{u-1}}} d_H([xG]_{u+m}, [x'G]_{u+m})$$

$d_H([y]_{u+m}, [y']_{u+m})$, where $d_H(\cdot, \cdot)$ denotes the Hamming distance between the two arguments and the minimization is over all \underline{x} and \underline{x}' with $\underline{x}_u \neq \underline{x}'_u$, $0 \leq u < \infty$. |

The second equality in definition 2.1 follows from the fact that $d_H([\underline{x} \underline{G}]_{u-1}, [\underline{x}' \underline{G}]_{u-1}) = 0$ if $[\underline{x}]_{u-1} = [\underline{x}']_{u-1}$, and hence the Hamming distance between codewords is not affected. The third equality follows from the fact that all pairs of codewords with $[\underline{x}]_{u-1} \neq [\underline{x}']_{u-1}$ were already included in the minimization for some smaller value of u and hence cannot change the minimum.

For periodic time-varying convolutional encoders with period T , definition 2.1 reduces to the following definition of d_{FD} .

Definition 2.2 $d_{FD} = \min_{0 \leq u < T} \min_{\substack{\underline{x}_u \neq \underline{x}'_u}} d_H([\underline{x} \underline{G}]_{u+m},$

$[\underline{x}' \underline{G}]_{u+m}) = \min_{0 \leq u < T} \min_{\substack{\underline{x}_u \neq \underline{x}'_u}} d_H([y]_{u+m}, [y']_{u+m}). |$

For time-varying encoders with finite memory $m < \infty$, the pattern of the first $K(m + 1)$ rows of \underline{G} must be repeated somewhere in \underline{G} . Suppose this pattern is repeated beginning at row KT (the $(KT + 1)^{st}$ row of \underline{G}), where T is some positive integer. Then a periodic encoder with period T can be formed

by reproducing periodically the first KT rows of \underline{G} . It can easily be seen from definitions 2.1 and 2.2 that d_{FD} for the periodic encoder is at least as great as d_{FD} for the original encoder. Similar arguments to the above can be given for each of the distance measures defined in this chapter. Hence as far as distance properties are concerned it is of no value to consider non-periodic time-varying encoders. Therefore all time-varying encoders will be considered to be periodic.

Finally, d_{FD} will now be defined for fixed encoders.

$$\text{Definition 2.3} \quad d_{FD} = \min_{\substack{\underline{x}_0 \neq \underline{x}'_0}} d_H([\underline{x} \underline{G}]_m, [\underline{x}' \underline{G}]_m) = \min_{\substack{\underline{x}_0 \neq \underline{x}'_0}} d_H([\underline{y}]_m, [\underline{y}']_m).$$

$$d_H([\underline{y}]_m, [\underline{y}']_m).$$

Because of the linearity of convolutional encoders, definitions 2.2 and 2.3 can be simplified.

Theorem 2.1 $d_{FD} = \min_{0 \leq u < T} \min_{\substack{\underline{x}_u \neq \underline{0}}} w_H([\underline{x} \underline{G}]_{u+m}) = \min_{0 \leq u < T} \min_{\substack{\underline{x}_u \neq \underline{0}}} w_H([\underline{y}]_{u+m})$, where $w_H(\cdot)$ denotes the Hamming weight of the argument.

Proof Let \underline{x} and \underline{x}' be any two information sequences with $\underline{x}_u \neq \underline{x}'_u$ such that $[\underline{y}]_{u+m} = [\underline{x} \underline{G}]_{u+m}$ and $[\underline{y}']_{u+m} = [\underline{x}' \underline{G}]_{u+m}$, and let $\underline{x}'' = \underline{x} + \underline{x}'$. Then $\underline{x}_u'' \neq \underline{0}$ and $[\underline{y}'']_{u+m} = [\underline{x}'' \underline{G}]_{u+m} = [(\underline{x} + \underline{x}') \underline{G}]_{u+m} = [\underline{x} \underline{G} + \underline{x}' \underline{G}]_{u+m} = [\underline{x} \underline{G}]_{u+m} + [\underline{x}' \underline{G}]_{u+m} = [\underline{y}]_{u+m} + [\underline{y}']_{u+m}$ since the encoder is linear. Therefore, $d_H([\underline{y}]_{u+m}, [\underline{y}']_{u+m}) = w_H([\underline{y}]_{u+m} + [\underline{y}']_{u+m}) = w_H([\underline{y}'']_{u+m})$

and the theorem follows immediately. |

For fixed encoders, Wozencraft and Reiffen [3] proved that

$$d_{FD} = \min_{\underline{x}_0 \neq \underline{0}} w_H(\underline{x} \underline{G})_m = \min_{\underline{x}_0 \neq \underline{0}} w_H(\underline{y})_m. \quad (62)$$

A generalization of definitions 2.2 and 2.3 will prove useful later in the construction of convolutional encoders with high d_{FD} .

Definition 2.4 The order j column distance, d_j , of a periodic encoder is given by

$$\begin{aligned} d_j &= \min_{0 \leq u < T} \min_{\underline{x}_u \neq \underline{x}'_u} d_H([\underline{x} \underline{G}]_u, [\underline{x}' \underline{G}]_{u+j}) = \min_{0 \leq u < T} \\ &\min_{\underline{x}_u \neq \underline{x}'_u} d_H([\underline{y}]_u, [\underline{y}']_{u+j}), \quad j = 0, 1, 2, \dots, \text{ and } d_\infty = \\ &\lim_{j \rightarrow \infty} d_j. \end{aligned}$$

Definition 2.5 The order j column distance, d_j , of a fixed encoder is given by

$$\begin{aligned} d_j &= \min_{\underline{x}_0 \neq \underline{x}'_0} d_H([\underline{x} \underline{G}]_j, [\underline{x}' \underline{G}]_j) = \min_{\underline{x}_0 \neq \underline{x}'_0} d_H([\underline{y}]_j, [\underline{y}']_j), \\ j &= 0, 1, 2, \dots, \text{ and } d_\infty = \lim_{j \rightarrow \infty} d_j. \end{aligned}$$

Some simple properties of the column distance will now be collected. The first property follows as a direct consequence of definitions 2.2 through 2.5, and applies to both periodic and fixed encoders.

Property Cl $d_{FD} = d_m$. |

The next property can be proved by a slight modification of the proof of theorem 2.1.

Property C2 $d_j = \min_{0 \leq u < T} \min_{\underline{x}_u \neq \underline{0}} w_H([\underline{x} \underline{G}]_{u+j}) = \min_{0 \leq u < T}$

$\min_{\underline{x}_u \neq \underline{0}} w_H([\underline{y}]_{u+j})$ for periodic encoders and $d_j = \min_{\underline{x}_0 \neq \underline{0}}$

$w_H([\underline{x} \underline{G}]_j) = \min_{\underline{x}_0 \neq \underline{0}} w_H([\underline{y}]_j)$ for fixed encoders,

$j = 0, 1, 2, \dots . |$

Let $\underline{g}_i = \begin{bmatrix} g_{0i}^{(1)} & g_{0i}^{(2)} & \dots & g_{0i}^{(N)} & g_{li}^{(1)} & g_{li}^{(2)} & \dots & g_{li}^{(N)} & \dots \end{bmatrix}$

be the i^{th} row of the generator matrix of a fixed encoder,

$1 \leq i \leq K$. \underline{g}_i contains N generator sequences and is called

the i^{th} generator. Then let $[\underline{g}_i]_j = \begin{bmatrix} g_{0i}^{(1)} & \dots & g_{0i}^{(N)} & \dots & g_{ji}^{(1)} & \dots & g_{ji}^{(N)} \end{bmatrix}$ be the $(j + 1)N$ -tuple consisting of the first $(j + 1)N$

entries in \underline{g}_i . This notation simplifies the statement of the next property for fixed encoders.

Property C3 $d_j \leq \min_{0 \leq u < T} \min_{1 \leq i \leq K} w_H([g_{0i}^{(1)}(u) \dots$

$g_{0i}^{(N)}(u) \dots g_{ji}^{(1)}(u) \dots g_{ji}^{(N)}(u)])$ for periodic encoders

and $d_j \leq \min_{1 \leq i \leq K} w_H([\underline{g}_i]_j)$ for fixed encoders, $j = 0, 1, 2, \dots . |$

Proof The proof will be given only for fixed encoders in order to simplify the notation. Let ℓ be the integer, $1 \leq \ell \leq K$, such that $w_H([\underline{g}_\ell]_j) \leq w_H([\underline{g}_i]_j)$, $1 \leq i \leq K$, $i \neq \ell$. Then let $\underline{x}_0 = [0 \dots 0 \ 1 \ 0 \dots 0]$, $\underline{x}_1 = \underline{x}_2 = \dots = \underline{0}$, where the 1 is in the ℓ^{th} position. Therefore $[\underline{x} \underline{G}]_j = [\underline{g}_\ell]_j$ and hence by property C2, $d_j \leq \min_{1 \leq i \leq K} w_H([\underline{g}_i]_j)$, $j = 0, 1, 2, \dots . |$

Property C4 $d_j \leq d_{j+1}$, $j = 0, 1, 2, \dots$, for both periodic and fixed encoders.

Proof $[y]_{u+j+1} = \left[[y]_{u+j}, y_{u+j+1} \right]$. But $\min_{0 \leq u < T}$

$$\min_{\substack{x_u \neq 0 \\ 0 \leq u < T}} w_H([y]_{u+j}) \leq \min_{0 \leq u < T} \min_{\substack{x_u \neq 0 \\ 0 \leq u < T}} w_H([y]_{u+j}, y_{u+j+1})$$

$$= \min_{0 \leq u < T} \min_{\substack{x_u \neq 0 \\ 0 \leq u < T}} w_H([y]_{u+j+1}). \text{ Therefore } d_j \leq d_{j+1},$$

$j = 0, 1, 2, \dots$. The proof for fixed encoders is similar and hence is omitted. |

B. Definite Decoding Minimum Distance

A definite decoding minimum distance, d_{DD} , and a definite decoding constraint length, n_{DD} , will be defined such that $\left\lfloor \frac{d_{DD} - 1}{2} \right\rfloor$ errors within a definite decoding constraint length are guaranteed correctable for codes being used with definite decoding. It is convenient to distinguish non-systematic encoders from systematic encoders.

1. Non-systematic Encoders

Robinson [7] suggested the following definitions of n_{DD} and d_{DD} . These definitions will be restricted to non-systematic encoders and the definition of d_{DD} will be generalized to periodic encoders.

Definition 2.6 The definite decoding constraint length n_{DD} is the number of error digits which can affect the syndrome equations from time u through time $u + \bar{m}$, $u \geq \bar{m}$, i.e., $n_{DD} = (2\bar{m} + 1)N$. |

Definition 2.7 $d_{DD} = \min_{2\bar{m} \leq u < 2\bar{m} + T} \min_{\substack{x_u \neq x_u'}}$

$$d_H([x \underline{G}]_{u-\bar{m}, u+\bar{m}}, [x' \underline{G}]_{u-\bar{m}, u+\bar{m}}) = \min_{2\bar{m} \leq u < 2\bar{m} + T}$$

$$\min_{\substack{x_u \neq x_u' \\ 2\bar{m} \leq u < 2\bar{m} + T}} d_H([y]_{u-\bar{m}, u+\bar{m}}, [y']_{u-\bar{m}, u+\bar{m}})$$

where the minimization is over all \underline{x} and \underline{x}' with $\underline{x}_u \neq \underline{x}'_u$,
 $2\bar{m} \leq u < 2\bar{m} + T$. |

For fixed encoders, definition 2.7 reduces to Robinson's [7] definition of d_{DD} .

$$\begin{aligned}\text{Definition 2.8} \quad d_{DD} &= \min_{\substack{\underline{x} \neq \underline{x}' \\ 2\bar{m} \leq u < 2\bar{m}}} d_H([\underline{x} G]_{\bar{m}, 3\bar{m}}, [\underline{x}' G]_{\bar{m}, 3\bar{m}}) \\ &= \min_{\substack{\underline{x} \neq \underline{x}' \\ 2\bar{m} \leq u < 2\bar{m}}} d_H([\underline{y}]_{\bar{m}, 3\bar{m}}, [\underline{y}']_{\bar{m}, 3\bar{m}}). |\end{aligned}$$

A similar argument to that used to prove theorem 2.1 results in the following simplification of definition 2.7.

$$\begin{aligned}\text{Theorem 2.2} \quad d_{DD} &= \min_{2\bar{m} \leq u < 2\bar{m} + T} \min_{\substack{\underline{x}_u \neq \underline{0} \\ u-\bar{m}, u+\bar{m}}} w_H([\underline{x} G]_{u-\bar{m}, u+\bar{m}}) \\ &= \min_{2\bar{m} \leq u < 2\bar{m} + T} \min_{\substack{\underline{x}_u \neq \underline{0} \\ u-\bar{m}, u+\bar{m}}} w_H([\underline{y}]_{u-\bar{m}, u+\bar{m}}). |\end{aligned}$$

Robinson [7] has also shown that for fixed encoders definition 2.8 can be simplified to

$$d_{DD} = \min_{\substack{\underline{x} \neq \underline{0} \\ 2\bar{m} \leq u < 2\bar{m}}} w_H([\underline{x} G]_{\bar{m}, 3\bar{m}}) = \min_{\substack{\underline{x} \neq \underline{0} \\ 2\bar{m} \leq u < 2\bar{m}}} w_H([\underline{y}]_{\bar{m}, 3\bar{m}}). \quad (63)$$

2. Systematic Encoders

For canonic systematic encoders the syndrome equations from time unit u through time unit $u + \bar{m}$ can be written as

$$[\underline{s}]_{u, u+\bar{m}} = [\underline{e}]_{u-\bar{m}, u+\bar{m}} \begin{bmatrix} Q_{\bar{m}}(u-\bar{m}) & & & \\ \cdots & \cdots & 0 & \\ 0_{N-K} & & \cdot & \\ \cdot & & \cdot & \\ \cdot & \cdots & Q_{\bar{m}}(u) & \\ \underline{Q}_{\bar{m}} & & & \\ \cdots & & \cdots & \\ I_{N-K} & \cdots & 0_{N-K} & \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ 0 & \cdots & Q_0(u+\bar{m}) & \\ & & \cdots & \\ & & I_{N-K} & \end{bmatrix} \quad (64)$$

where each $Q_i(u)$, $0 \leq i \leq \bar{m}$, $0 \leq u < \infty$, is a $K \times (N - K)$ matrix of elements from $GF(q)$. Hence the last $(N - K)$ digits in $\underline{e}_{u-\bar{m}}$, $\underline{e}_{u-\bar{m}+1}$, ..., \underline{e}_{u-2} , and \underline{e}_{u-1} , i.e., the digits in the parity positions of all the error blocks previous to time u , do not affect the syndrome equations. Therefore definitions 2.6, 2.7, and 2.8 can be modified for systematic encoders as suggested by Massey [8].

Definition 2.9 n_{DD} , the number of error digits which can affect the syndrome equations from time u through time $u + \bar{m}$, $u \geq \bar{m}$, is $N(\bar{m} + 1) + K\bar{m}$ for systematic encoders. |

Definition 2.10 $d_{DD} = \min_{2\bar{m} \leq u < 2\bar{m} + T} \min_{\underline{x}_u \neq \underline{x}'_u} d_H([\underline{x}]_{u-\bar{m}, u-1}, [\underline{x} G]_{u, u+\bar{m}}), [\underline{x}']_{u-\bar{m}, u-1}, [\underline{x}' G]_{u, u+\bar{m}})$

for periodic systematic encoders. |

Definition 2.11 $d_{DD} = \min_{\substack{\underline{x}_{2m} \neq \underline{x}'_{2m}}} d_H(\left[\begin{smallmatrix} \underline{x} \\ m, 2m-1 \end{smallmatrix} \right], \left[\begin{smallmatrix} \underline{x} \\ 2m, 3m \end{smallmatrix} \right], \left[\begin{smallmatrix} \underline{x}' \\ m, 2m-1 \end{smallmatrix} \right], \left[\begin{smallmatrix} \underline{x}' \\ 2m, 3m \end{smallmatrix} \right])$ for fixed systematic encoders.

Again, a similar argument to that used to prove theorems 2.1 and 2.2 results in the following simplification of definition 2.10.

Theorem 2.3 $d_{DD} = \min_{2m \leq u < 2m+T} \min_{\substack{\underline{x}_u \neq 0}} w_H(\left[\begin{smallmatrix} \underline{x} \\ u-m, u-1 \end{smallmatrix} \right], \left[\begin{smallmatrix} \underline{x} \\ u, u+m \end{smallmatrix} \right]).$

For fixed encoders, Massey [8] has shown that definition 2.11 can also be simplified to

$$d_{DD} = \min_{\substack{\underline{x}_{2m} \neq 0}} w_H(\left[\begin{smallmatrix} \underline{x} \\ m, 2m-1 \end{smallmatrix} \right], \left[\begin{smallmatrix} \underline{x} \\ 2m, 3m \end{smallmatrix} \right]). \quad (65)$$

As discussed in Chapter I, if the decoding memory is chosen according to equation (52) for periodic systematic encoders or equation (53) for fixed systematic encoders, then the decoding memory is the same as the encoding memory, and \bar{m} can be replaced by m in definitions 2.9 through 2.11, in theorem 2.3, and in equation (65).

It can easily be seen from definitions 2.2, 2.7, and 2.10 that for periodic encoders

$$d_{DD} \leq d_{FD}, \quad (66)$$

a result which is well known [7] for fixed encoders.

For each definition of d_{DD} in Section II.B, an argument similar to Robinson's [7] for fixed non-systematic encoders can be used to show that $\left\lfloor \frac{d_{DD} - 1}{2} \right\rfloor$ errors within a definite decoding constraint length are guaranteed correctable by an algebraic decoder operating in the definite decoding mode.

C. Free Distance

For decoding schemes such as sequential decoding, in which the decoder is not constrained to consider only one constraint length of received digits while attempting to decode a particular block of transmitted digits, but may search over a much longer portion of the received sequence, d_{FD} and d_{DD} are no longer meaningful. Consequently a different distance measure, called the free distance by Massey [9], and studied by McEliece and Rumsey [10] and Neumann [11], will be considered. Free distance is defined over the entire encoded sequence and hence is appropriate for a decoder which makes its decisions on the basis of the entire received sequence.

Before defining free distance, a more general distance measure will be introduced.

Definition 2.12 The order j row distance, r_j , of a periodic encoder is given by

$$r_j = \min_{0 \leq u < T} \min_{\substack{x_u \neq x'_u \\ x_u = x'_{u-1}}} d_H \left(\left[\begin{matrix} x \\ \vdots \\ x_{u+j} \end{matrix} \right], \left[\begin{matrix} x' \\ \vdots \\ x'_{u+j} \end{matrix} \right] \right) ,$$

$$\left[\begin{matrix} x \\ \vdots \\ x_{u-1} \end{matrix} \right] = \left[\begin{matrix} x' \\ \vdots \\ x'_{u-1} \end{matrix} \right]$$

$j = 0, 1, 2, \dots$, and $r_\infty = \lim_{j \rightarrow \infty} r_j$, where $\underline{0}^\infty$ represents an infinite concatenation of all-zero K-tuples.]

Definition 2.13 The order j row distance, r_j , of a fixed encoder is given by

$$r_j = \min_{\underline{x}_0 \neq \underline{x}'_0} d_H \left(\begin{bmatrix} \underline{x} \\ \underline{x}' \end{bmatrix}_j, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \right) \text{G}, \quad j = 0, 1, 2, \dots, \text{ and } r_\infty = \lim_{r \rightarrow \infty} r_j.$$

Some simple properties of the row distance will now be given. The first property can be proved by again slightly modifying the proof of theorem 2.1

Property R1 $r_j = \min_{0 \leq u < T} \min_{\substack{\underline{x}_u \neq \underline{0} \\ [\underline{x}]_{u-1} = \underline{0}}} w_H \left(\begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}_{u+j}, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \right) \text{G}$

for periodic encoders and $r_j = \min_{\underline{x}_0 \neq \underline{0}} w_H \left(\begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}_j, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \right) \text{G}$ for fixed encoders, $j = 0, 1, 2, \dots$.

As was seen in property C4, the column distance d_j cannot decrease with increasing j . Just the opposite is true for row distance, as will now be shown.

Property R2 $r_j \geq r_{j+1}$, $j = 0, 1, 2, \dots$, for both periodic and fixed encoders.

Proof Let $Y = \left\{ \begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}_{u+j+1}, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \mid [\underline{x}]_{u-1} = \underline{0}, \underline{x}_u \neq \underline{0} \right\}$ and $Y' = \left\{ \begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}_{u+j}, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \mid [\underline{x}]_{u-1} = \underline{0}, \underline{x}_u \neq \underline{0} \right\}$. Then Y contains $2^{K(j+1)} (2^K - 1)$ elements and Y' contains $2^{Kj} (2^K - 1)$ elements. Clearly $Y' \subset Y$. Therefore $\min_{\substack{\underline{x}_u \neq \underline{0} \\ [\underline{x}]_{u-1} = \underline{0}}} w_H \left(\begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}_{u+j+1}, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \right) \leq \min_{\substack{\underline{x}_u \neq \underline{0} \\ [\underline{x}]_{u-1} = \underline{0}}} w_H \left(\begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}_{u+j}, \begin{bmatrix} \underline{0}^\infty \\ \underline{0}^\infty \end{bmatrix} \right) \text{G}$. Hence $r_{j+1} \leq r_j$,

$j = 0, 1, 2, \dots$, for periodic encoders. The proof for fixed encoders follows in a similar fashion. |

Property R3 $r_j \leq \min_{0 \leq u < T} \min_{1 \leq i \leq K} w_H \left[g_{0i}^{(1)}(u) \dots g_{0i}^{(N)}(u) g_{li}^{(1)}(u) \dots g_{li}^{(N)}(u) \dots \right]$ for periodic encoders and $r_j \leq \min_{1 \leq i \leq K} w_H(g_i)$ for fixed encoders, $j = 0, 1, 2, \dots$.

Proof The proof will be given only for fixed encoders in order to simplify the notation. $r_0 = \min_{1 \leq i \leq K} w_H(g_i)$. Therefore property R2 implies that $r_j \leq r_0 = \min_{1 \leq i \leq K} w_H(g_i)$, $j = 0, 1, 2, \dots$. |

Now a simple result relating the column distance d_j to the row distance r_j will be given.

Theorem 2.4 $d_0 \leq d_1 \leq \dots \leq d_\infty \leq r_\infty \leq \dots \leq r_1 \leq r_0$ for both periodic and fixed encoders.

Proof Again the proof will be given only for fixed encoders in order to simplify the notation involved. Let $\underline{y} = \left[[\underline{x}]_j, \underline{\varrho}^\infty \right] \underline{G}$ and $\underline{y}' = [\underline{x} \underline{G}]_j$. Then $\underline{y} = \left[\underline{y}', \underline{y}_{j+1}, \dots, \underline{y}_{j+m} \right]'$, $j = 0, 1, 2, \dots$. Hence $w_H(\underline{y}') \leq w_H(\underline{y})$ and $r_\infty = \lim_{j \rightarrow \infty} r_j = \lim_{j \rightarrow \infty} \min_{\underline{x}_0 \neq \underline{\varrho}} w_H([\underline{x}]_j, \underline{\varrho}^\infty) \underline{G} \geq \lim_{j \rightarrow \infty} \min_{\underline{x}_0 \neq \underline{\varrho}} w_H([\underline{x} \underline{G}]_j) = \lim_{j \rightarrow \infty} d_j = d_\infty$. Clearly property C4 implies that $d_j \leq \lim_{j' \rightarrow \infty} d_{j'} = d_\infty$ and property R2 implies that $r_j \geq \lim_{j' \rightarrow \infty} r_{j'} = r_\infty$ for all finite j . The theorem then follows immediately from these two properties. |

Now the free distance, d_{FREE} , will be defined as the minimum weight encoded sequence such that $\underline{x} \neq \underline{0}$.

Definition 2.14 $d_{\text{FREE}} = \min_{\substack{\underline{x} \neq \underline{0}}} w_H(\underline{x} \underline{G})$ for both periodic and fixed encoders.

Note that d_{FREE} is a property of the code itself.

Some properties of d_{FREE} will be presented next.

Property F1 $d_{\text{FREE}} = d_{\infty}$ for both fixed and periodic encoders.

Proof For fixed encoders, $d_{\infty} = \lim_{j \rightarrow \infty} \min_{\substack{\underline{x}_0 \neq \underline{0}}} w_H([\underline{x} \underline{G}]_j) = \min_{\substack{\underline{x}_0 \neq \underline{0}}} w_H(\underline{x} \underline{G})$. Now let n be the smallest integer such that $\underline{x}_n \neq \underline{0}$. Then let \underline{x}' be the information sequence whose transform is $\underline{x}'(D) = D^{-n} \underline{x}(D)$, where $\underline{x}(D)$ is the transform of \underline{x} . Clearly $\underline{x}' \neq \underline{0}$, \underline{x}' is causal, and $w_H(\underline{x}' \underline{G}) = w_H(\underline{x} \underline{G})$. Hence every codeword produced by a non-zero input sequence \underline{x} has the same weight as the codeword produced by the input sequence \underline{x}' with $\underline{x}'_0 \neq \underline{0}$. Therefore $\min_{\substack{\underline{x}_0 \neq \underline{0}}} w_H(\underline{x} \underline{G}) = \min_{\substack{\underline{x} \neq \underline{0}}} w_H(\underline{x} \underline{G})$ and

$d_{\text{FREE}} = d_{\infty}$. The proof for periodic encoders follows in a similar fashion.

The second property follows directly from property C1, theorem 2.4, property F1, and definition 2.14.

Property F2 $d_{\text{FD}} \leq d_{\text{FREE}} \leq \min_{0 \leq u < T} \min_{1 \leq i \leq k} w_H([g_{0i}^{(1)} \dots g_{0i}^{(N)} \dots g_{li}^{(1)} \dots g_{li}^{(N)} \dots]^{(u)})$ for periodic encoders and $d_{\text{FD}} \leq d_{\text{FREE}} \leq \min_{1 \leq i \leq k} w_H(g_i)$ for fixed encoders.

The third property shows that d_{FREE} is unchanged if the set of allowable input sequences is expanded to include all rational sequences. Hence this property serves as an alternate definition of d_{FREE} .

Property F3 $d_{\text{FREE}} = \min_{\substack{\underline{x} \neq \underline{0}}} w_H(\underline{y})$, where the minimization

is over all rational input sequences $\underline{x} \neq \underline{0}$, for both periodic and fixed encoders.

Proof Let \underline{x} be any non-causal rational input sequence and let n be the least integer such that $\underline{x}_n \neq \underline{0}$. Then let \underline{x}' be the input sequence whose transform is $\underline{x}'(D) = D^{kT} \underline{x}(D)$, where $\underline{x}(D)$ is the transform of \underline{x} , T is the period of the encoder, and k is the least positive integer such that $kT \geq -n$.

Clearly \underline{x}' is causal, $\underline{x}' \neq \underline{0}$, and $w_H(\underline{y}') = w_H(\underline{y})$, where \underline{y}' is the codeword produced by \underline{x}' and \underline{y} is the codeword produced by \underline{x} . Hence every codeword produced by a non-causal input sequence \underline{x} has the same weight as the codeword produced by the causal input sequence \underline{x}' . Therefore the set of input sequences over which the Hamming weight of codewords is minimized in the definition of d_{FREE} can be expanded to include all rational sequences without changing d_{FREE} . |

The fourth property applies only to fixed polynomial encoders, i.e., to fixed encoders whose matrix of generator functions $\underline{G}(D)$ contains only polynomial elements.

Property F4 For all fixed polynomial encoders with a feed-forward inverse, $d_{\text{FREE}} = d_j = d_{j+1} = \dots = d_\infty = r_\infty = \dots = r_{j-m}$ for some finite j .

Proof A fixed polynomial encoder has a feedforward inverse if there exists an $N \times K$ matrix of polynomials $\underline{G}^{-1}(D)$ such that $\underline{G}(D) \underline{G}^{-1}(D) = D^L I_K$, where L is called the delay of the

inverse. Massey and Sain [12] have shown that $L \leq Km$, where m is the memory of the encoder. Necessary and sufficient conditions for a fixed encoder to have a feedforward (i.e., polynomial) inverse are given by Olson [13]. Clearly, if $\underline{G}(D) \underline{G}^{-1}(D) = D^L I_K$, then $\underline{x}(D) \underline{G}(D) \underline{G}^{-1}(D) = \underline{y}(D) \underline{G}^{-1}(D) = D^L \underline{x}(D)$ for any input sequence whose transform is $\underline{x}(D)$.

In minimizing $w_H(\underline{x} \underline{G})$ over all input sequences $\underline{x} \neq \underline{0}$, it is not necessary to consider any input sequence with a string of m or more blocks of $\underline{0}$'s in it since following such a string with additional non-zero blocks can only add to the weight of the codeword. Also, property F2 implies that d_{FREE} can never be more than $N(m + 1)$, the maximum number of 1's in any generator, i.e., d_{FREE} is finite.

Now let M be the maximum degree of the polynomials in $\underline{G}^{-1}(D)$. Clearly, if \underline{y} contains a string of $(m + M + 1)$ or more blocks of $\underline{0}$'s, \underline{x} must contain a string of m or more blocks of $\underline{0}$'s. Hence for each input sequence \underline{x} with $\underline{x}_0 \neq \underline{0}$ which is capable of producing the minimum free weight codeword, $\underline{y}_L \neq \underline{0}$ and each successive string of $(m + M + 1)$ blocks of encoded digits must contain at least one non-zero block. Therefore the first $(Nm + N - 1)(M + m + 1) + L + 1$ blocks of encoded digits must have weight at least $N(m + 1)$ for all encoded sequences capable of producing the minimum free weight codeword. This implies that $d_{\text{FREE}} = d_\infty = \min_{\substack{\underline{x}_0 \neq \underline{0}}} w_H(\underline{x} \underline{G}) = d_{(Nm + N - 1)(M + m + 1) + L}$ and that $d_{\text{FREE}} = d_\infty = \min_{\substack{\underline{x}_0 \neq \underline{0}}} w_H(\underline{x} \underline{G}) = r_{(Nm + N - 1)(M + m + 1) + L - m}$. Hence from

theorem 2.4, $d_{\text{FREE}} = d_j = d_{j+1} = \dots = d_\infty = r_\infty = \dots = r_{j-m}$
for some finite $j.$

Property F4 is a very crude bound on the length of information sequence needed to produce the minimum free weight codeword. It is conjectured that this result can be strengthened considerably by more detailed arguments. Neumann [11] has suggested that the true bound is m , i.e., that $d_{\text{FREE}} = r_m$. However, as will be shown in Chapter V, this is not the case in general although it may be true for $R = 1/2$ systematic fixed encoders.

Example 2.1 Consider an $R = 1/2$ fixed polynomial encoder with

$$\underline{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots \\ & 1 & 1 & 1 & 1 & 0 & 0 & \dots \\ & & \ddots & & & & & \end{bmatrix} . \quad \text{Clearly } d_0 = d_1 = \dots = d_\infty = \\ d_{\text{FREE}} = 2 \text{ and } r_0 = r_1 = \dots = r_\infty = 4. \text{ Since } d_\infty \neq r_\infty, \text{ this encoder does not have a feedforward inverse.}$$

Consider an algebraic decoder which decodes in "frames" of $\ell + 1$ blocks of received digits, $\ell \gg m$, and which decodes each received sequence $[\underline{r}]_\ell$ into the encoded sequence $[\underline{y}]_\ell$ such that $d_H([\underline{r}]_\ell, [\underline{y}]_\ell)$ is minimum. A feedback decoder operating over such a frame will make a decoding error for at least one pattern of $\left\lfloor \frac{d_{FD} + 1}{2} \right\rfloor$ errors in the frame whereas the above algebraic decoder cannot make an error unless $\left\lfloor \frac{r_{\ell-m} + 1}{2} \right\rfloor$ errors occur in the frame.

Now consider the binary symmetric channel (BSC) shown in Figure 2.1. p is the digit error probability. Let q be the probability that e digits were received incorrectly in n successive uses of the channel. Clearly

$$q = p^e (1 - p)^{n-e} . \quad (67)$$

Now let q' be the probability that $e + 1$ digits were received incorrectly in n successive uses of the channel. Then

$$q' = p^{e+1} (1 - p)^{n-e-1} = \frac{qp}{1 - p} . \quad (68)$$

If p is very small, i.e., $p \ll 1$, then

$$q' \ll q . \quad (69)$$

Therefore if p is small enough it can be seen that the decoding error probability is a function only of the minimum number of incorrectly received digits in the frame that can cause a decoding error, since heavier error patterns occur with negligible probability. Since in general $r_{\ell-m} > d_{FD}$, the decoding error probability is lower for the decoder operating over the whole frame than for the feedback decoder.

Therefore $r_{\ell-m}$ is the appropriate distance measure for the decoder which decodes in frames of $\ell + 1$ received blocks since it determines the decoding error probability. Property F4 implies that if ℓ is large enough, $r_{\ell-m} = d_{FREE}$ and d_{FREE} becomes the appropriate distance measure for the decoder which decodes in frames of $\ell + 1$ received blocks.

A sequential decoder, although not algebraic, does decode in frames of many received blocks. Hence d_{FREE} would

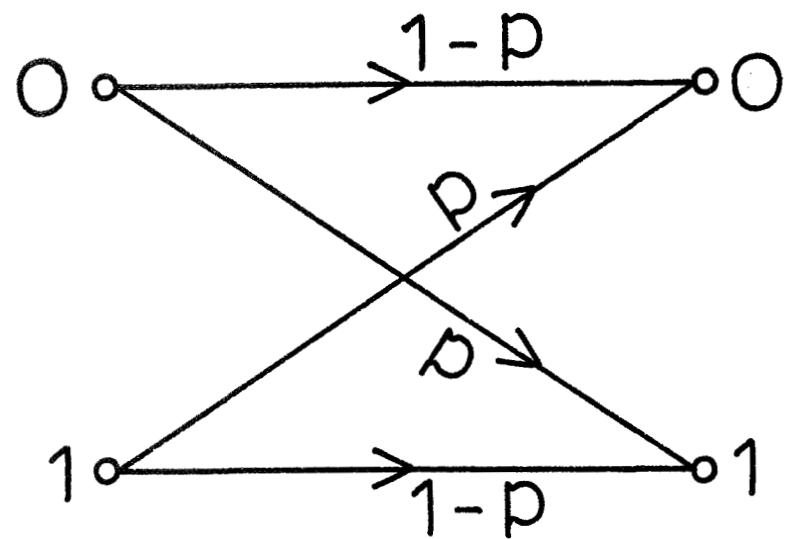


Fig. 2.1. A binary symmetric channel.

seem to be a more appropriate distance measure than either d_{FD} or d_{DD} for encoders used with sequential decoding. Chapter VI presents simulation results which corroborate this reasoning.

D. Reverse Distance

Consider an encoder G with generator matrix

$$\underline{G} = \begin{bmatrix} G_0(0) & G_1(0) & \dots & G_m(0) & & \underline{0} \dots \\ 0 & G_0(1) & \dots & G_{m-1}(1) & G_m(1) & \dots \end{bmatrix} . \quad (70)$$

Then for any of the distance measures defined in this chapter, the following definition holds.

Definition 2.15 The reverse distance of the encoder G is the distance of the encoder G' with generator matrix

$$\underline{G}' = \begin{bmatrix} G_m(0) & G_{m-1}(0) & \dots & G_0(0) & & \underline{0} \dots \\ 0 & G_m(1) & \dots & G_1(1) & G_0(1) & \dots \end{bmatrix} .$$

If $\underline{G}' = \underline{G}$, then the encoder G is called reversible. Reversible encoders have been studied by Massey [14] and Robinson [15].

Reverse distance will prove useful in Chapter V, when some methods for calculating d_{FREE} are derived.

III. Encoder Equivalence and Syndrome Formation

A. Encoder Equivalence

There is considerable interest in finding systematic equivalents for non-systematic encoders for the following reasons: (1) systematic encoders are in general simpler to implement than non-systematic encoders; (2) the encoded sequences from systematic encoders possess the "quick look" property, i.e., the noisy version of the information sequence is directly available at the receiver.

As will be shown in later chapters, for a given rate and a given constraint length, non-systematic encoders are superior to systematic encoders for sequential decoding because larger free distances are achievable. For some algebraic decoding techniques, however, such as feedback decoding, where column distance is the important parameter, non-systematic encoders are no longer superior and it is usually desirable to use only systematic encoders.

1. Rational Equivalence

Definition 3.1 Two encoders are rationally equivalent if they have the same set of output sequences over the set of all rational input sequences. |

Consider a fixed non-systematic encoder with a matrix of generator functions $\underline{G}(D)$. Then

$$\underline{y}(D) = \underline{x}(D) \underline{G}(D) \quad (71)$$

$$= \underline{x}(D) \underline{R}(D) \underline{R}^{-1}(D) \underline{G}(D) , \quad (72)$$

where $\underline{R}^{-1}(D)$ is the $K \times K$ inverse matrix of one of the non-singular $K \times K$ submatrices of $\underline{G}(D)$. (The definition of a convolutional encoder guarantees that some $K \times K$ submatrix of $\underline{G}(D)$ is non-singular.)

Therefore

$$\underline{y}(D) = \hat{\underline{x}}(D) \hat{\underline{G}}(D) , \quad (73)$$

where $\hat{\underline{x}}(D) = \underline{x}(D) \underline{R}(D)$, $\hat{\underline{G}}(D) = \underline{R}^{-1}(D) \underline{G}(D)$, and $\hat{\underline{G}}(D)$ is in systematic form. Hence if $\hat{\underline{G}}(D)$ is realizable, it is a systematic rational equivalent of $\underline{G}(D)$. However, in general there is no guarantee that $\hat{\underline{G}}(D)$ be realizable.

Since $\underline{R}(D)$ is just a $K \times K$ submatrix of $\underline{G}(D)$, it contains only realizable functions. Therefore $\underline{R}^{-1}(D)$, the matrix of cofactors of $\underline{R}(D)$ divided by $\det[\underline{R}(D)]$, contains only realizable functions if $\det[\underline{R}(D)]$ has a non-zero constant term. Hence a sufficient condition for $\hat{\underline{G}}(D)$ to be realizable is that $\det[\underline{R}(D)]$ have a non-zero constant term.

Example 3.1 Consider the $R = 2/3$ fixed binary non-systematic encoder with $\underline{G}(D) = \begin{bmatrix} 1 & 1+D & D \\ D & 1+D^2 & 1 \end{bmatrix}$. Choose $\underline{R}(D) = \begin{bmatrix} 1 & 1+D \\ D & 1+D^2 \end{bmatrix}$. Then $\det[\underline{R}(D)] = 1+D$ and $\underline{R}^{-1}(D) = \begin{bmatrix} 1+D & 1 \\ D/1+D & 1/1+D \end{bmatrix}$. Hence $\hat{\underline{G}}(D) = \begin{bmatrix} 1 & 0 & 1+D+D^2 \\ 0 & 1 & 1+D \end{bmatrix}$ is a systematic rational equivalent of $\underline{G}(D)$. |

Example 3.2 Consider the $R = 2/3$ fixed binary non-systematic encoder with $\underline{G}(D) = \begin{bmatrix} 1 & 1+D & D \\ D & D & D \end{bmatrix}$. Choose $\underline{R}(D) = \begin{bmatrix} 1 & 1+D \\ D & D \end{bmatrix}$. Then $\det[\underline{R}(D)] = D^2$ and $\underline{R}^{-1}(D) = \begin{bmatrix} 1/D & 1+D/D^2 \\ 1/D & 1/D^2 \end{bmatrix}$. Hence

$\hat{G}(D) = \begin{bmatrix} 1 & 0 & 1/D \\ 0 & 1 & 1+D/D \end{bmatrix}$ is not realizable. But if one chooses $R(D) = \begin{bmatrix} 1+D & D \\ D & D \end{bmatrix}$, then $\det[R(D)] = D$ and $R^{-1}(D) = \begin{bmatrix} 1 & 1 \\ 1 & 1+D/D \end{bmatrix}$.

In this case $\hat{G}(D) = \begin{bmatrix} 1+D & 1 & 0 \\ D & 0 & 1 \end{bmatrix}$ is in systematic form and is realizable even though $\det[R(D)]$ has a zero constant term. Hence this $\hat{G}(D)$ is a systematic rational equivalent of $G(D)$. |

In example 3.2, note that $d_1 = 2$ for the encoder with $\hat{G}(D) = \begin{bmatrix} 1+D & 1 & 0 \\ D & 0 & 1 \end{bmatrix}$ while the information sequence $[\underline{x}]_1 = [01, 10]$ produces a codeword $[\underline{y}]_1 = [000, 001]$ with weight 1 for the rationally equivalent encoder with $G(D) = \begin{bmatrix} 1 & 1+D & D \\ D & D & D \end{bmatrix}$, so that $d_1 = 1$ for this code. Hence rational equivalence does not necessarily imply equivalence of column distances.

2. Causal Equivalence

Definition 3.2 The set of causally driven output sequences of the encoder G , $(CDOS)_G$, is the set of all output sequences produced by causal input sequences. |

Definition 3.3 Two encoders G and \hat{G} are causally equivalent if they have the same set of causally driven output sequences, i.e., if $(CDOS)_G = (CDOS)_{\hat{G}}$. |

Lemma 3.1 For two fixed encoders G and \hat{G} , if $(CDOS)_G = (CDOS)_{\hat{G}}$, then for any two causal information sequences \underline{x} and $\hat{\underline{x}}$ such that $\underline{x} G = \hat{\underline{x}} \hat{G}$, $\underline{x}_0 \neq 0$ if and only if $\hat{\underline{x}}_0 \neq 0$.

Proof Assume $\underline{x}_0 = \underline{0}$, $\hat{\underline{x}}_0 \neq \underline{0}$. Let n be the least positive integer such that $\underline{x}_n \neq \underline{0}$. Then the information sequence whose transform is $D^{-n}\underline{x}(D)$ is causal and produces the output sequence whose transform is $\underline{y}(D) = D^{-n}\underline{x}(D)\underline{G}(D)$. Therefore $\underline{y}(D) = D^{-n}\hat{\underline{x}}(D)\hat{\underline{G}}(D)$. Since the definition of an encoder implies that no two information sequences can produce the same output sequence, $D^{-n}\hat{\underline{x}}(D)$ is the transform of the only input sequence to $\hat{\underline{G}}$ which can produce \underline{y} . But the sequence whose transform is $D^{-n}\hat{\underline{x}}(D)$ is not causal, contradicting the causal equivalence of G and \hat{G} . The only if part of the proof follows exactly as above. |

For a fixed encoder G , assume $\det[\underline{R}(D)]$ has a non-zero constant term, i.e., $\underline{R}^{-1}(D)$ is realizable. Then if $\hat{\underline{x}}(D)$ is causal, $\underline{x}(D) = \hat{\underline{x}}(D)\underline{R}^{-1}(D)$ is causal and it follows from equations (71) and (73) that

$$(CDOS)_G^A \subset (CDOS)_G . \quad (74)$$

Also if $\underline{x}(D)$ is causal, then $\hat{\underline{x}}(D) = \underline{x}(D)\underline{R}(D)$ is causal and

$$(CDOS)_G \subset (CDOS)_G^A . \quad (75)$$

Hence

$$(CDOS)_G = (CDOS)_G^A \quad (76)$$

and the two encoders have the same set of causally driven output sequences. Therefore lemma 3.1 implies that

$$d_j = \hat{d}_j, \quad j = 0, 1, 2, \dots, \quad (77)$$

where d_j is the order j column distance of G and \hat{d}_j is the order j column distance of \hat{G} . Theorem 3.1 summarizes the above results.

Theorem 3.1 Let $\underline{R}(D)$ be a $K \times K$ matrix formed with K columns of $\underline{G}(D)$. If $\det[\underline{R}(D)]$ has a non-zero constant term, then $\hat{\underline{G}}(D) = \underline{R}^{-1}(D) \underline{G}(D)$ is a systematic causal equivalent of $\underline{G}(D)$ and $d_j = \hat{d}_j$, $j = 0, 1, 2, \dots$.

If $\det[\underline{R}(D)]$ has a zero constant term for all $\binom{N}{K}$ possible choices of $\underline{R}(D)$, a systematic causal equivalent may not exist. In this case, as will next be shown, a rationally equivalent encoder \tilde{G} can be found such that $d_j \leq \tilde{d}_j$, $j = 0, 1, 2, \dots$, and such that \tilde{G} has a systematic causal equivalent. The following procedure will produce the encoder \tilde{G} . (Assume the numerators and denominators of the generator functions of G are relatively prime polynomials.)

Step 1 Convert $\underline{G}(D)$ to a polynomial matrix $\underline{G}'(D)$ by multiplying each row of $\underline{G}(D)$ by the least common multiple of the denominators of its generator functions.

Clearly this step preserves rational equivalence.

$$\underline{G}'(D) = \begin{bmatrix} L_1(D) & 0 & \dots & 0 \\ 0 & L_2(D) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & L_K(D) \end{bmatrix} \underline{G}(D) \quad , \quad (78)$$

where $L_i(D)$ is the least common multiple of the denominators of the generator functions in row i , $1 \leq i \leq K$. Let

$$\underline{L}(D) = \begin{bmatrix} L_1(D) & \dots & 0 \\ \vdots & \ddots & \\ \vdots & \vdots & \\ 0 & \dots & L_K(D) \end{bmatrix} . \quad (79)$$

Then

$$\underline{L}^{-1}(D) = \begin{bmatrix} 1/L_1(D) & \dots & 0 \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1/L_K(D) \end{bmatrix} \quad (80)$$

is realizable since $\underline{G}(D)$ realizable implies that each $L_i(D)$ has a non-zero constant term. Therefore if $\underline{x}'(D)$ is causal, then $\underline{x}(D) = \underline{x}'(D) \underline{L}(D) = [x^{(1)'}(D) L_1(D), \dots, x^{(K)'}(D) L_K(D)]$ is causal. Also if $\underline{x}(D)$ is causal, then $\underline{x}'(D) = \underline{x}(D) \underline{L}^{-1}(D) = [x^{(1)}(D)/L_1(D), \dots, x^{(K)}(D)/L_K(D)]$ is causal. Hence step 1 preserves causal equivalence.

Note that since $\det[\underline{R}(D)]$ is assumed to have a zero constant term, $\det[\underline{R}_0] = 0$ for all $\binom{N}{K}$ possible choices of $\underline{R}(D)$, where \underline{R}_0 is the $K \times K$ matrix of constant terms of $\underline{R}(D)$. Therefore $\text{rank}[\underline{G}_0] < K$ and since $\underline{G}_0' = I_K \underline{G}_0$, $\text{rank}[\underline{G}_0'] < K$.

Step 2 Convert $\underline{G}'(D)$ to a matrix $\underline{G}''(D)$ in which the K^{th} row of \underline{G}_0'' , i.e., the constant terms in the K^{th} row of $\underline{G}''(D)$, is all-zero by rearranging the rows of $\underline{G}'(D)$ and then adding a linear scalar combination of the first $(K-1)$ rows of $\underline{G}'(D)$ to row K . |

Again this step clearly preserves rational equivalence.

After rearranging the rows of $\underline{G}'(D)$,

$$\underline{G}''(D) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ c_1 & c_2 & \dots & c_K \end{bmatrix} \quad \underline{G}'(D) , \quad (81)$$

where each $c_i \in GF(2)$ and $c_K \neq 0$. Let

$$\underline{C}(D) = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ c_1 & \dots & c_K \end{bmatrix} . \quad (82)$$

Clearly $\underline{C}(D)$ is non-singular and the proof that causal equivalence is preserved is similar to the proof in step 1.

Step 3 Convert $\underline{G}''(D)$ to a matrix $\underline{G}'''(D)$ by multiplying row K of $\underline{G}''(D)$ by D^{-1} .

Since the row space of $\underline{G}''(D)$ over all rational input sequences is not changed by multiplying the Kth row of $\underline{G}''(D)$ by D^{-1} , $\underline{G}'''(D)$ is rationally equivalent to $\underline{G}''(D)$.

$$\underline{G}'''(D) = \underline{E}(D) \underline{G}''(D), \quad (83)$$

where

$$\underline{E}(D) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & D^{-1} \end{bmatrix} \quad (84)$$

is non-singular and non-realizable. Note that

$$\underline{E}^{-1}(D) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & D \end{bmatrix} \quad (85)$$

is realizable. Therefore if $\underline{x}''(D)$ is causal, then $\underline{x}'''(D) = \underline{x}''(D) \underline{E}^{-1}(D)$ is causal and

$$(CDOS)_{G''} \subset (CDOS)_{G'''}. \quad (86)$$

However, the encoders G'' and G''' are not in general causally equivalent, since $\underline{x}'''(D)$ causal does not imply that $\underline{x}''(D) = \underline{x}'''(D)$ $\underline{E}(D)$ is causal. Therefore the (CDOS) set is in general enlarged in step 3, and the encoder G''' is said to be causally dominant to the encoder G'' . It remains to show that $d_j''' \geq d_j'', j = 0, 1, 2, \dots$, where d_j''' is the order j column distance of G''' and d_j'' is the order j column distance of G'' .

Case 1 Let $\underline{x}'''(D)$ be the transform of a causal input sequence with $\underline{x}_0''' \neq \underline{0}$, $x_0^{(K)'''}=0$. Then $\underline{x}''(D) = \underline{x}'''(D)$ $\underline{E}(D)$ is causal and has $\underline{x}_0'' \neq \underline{0}$. Hence $\underline{y}''(D) = \underline{x}''(D)$ $\underline{G}''(D) = \underline{x}'''(D)$ $\underline{G}'''(D) = \underline{y}'''(D)$ and $w_H(\underline{y}'')_j = w_H(\underline{y}''')_j, j = 0, 1, 2, \dots . |$

Case 2 Let $\underline{x}'''(D)$ be the transform of a causal input sequence with $x_0^{(K)'''} \neq 0$ and let $\underline{y}'''(D) = \underline{x}'''(D)$ $\underline{G}'''(D)$. Then $\underline{\bar{x}}''(D) = D\underline{x}''(D) = D\underline{x}'''(D)$ $\underline{E}(D) = [Dx^{(1)'''}(D), \dots, Dx^{(K-1)'''}(D), x^{(K)'''}(D)]$ is causal, has $\underline{\bar{x}}_0'' \neq \underline{0}$, and produces an output sequence whose transform is $\underline{\bar{y}}''(D) = \underline{\bar{x}}''(D)$ $\underline{G}''(D) = \underline{\bar{x}}''(D)$ $\underline{E}^{-1}(D)$ $\underline{G}'''(D) = D\underline{y}'''(D)$. Clearly $w_H(\underline{\bar{y}}'')_j \leq w_H(\underline{y}''')_j, j = 0, 1, 2, \dots . |$

Cases 1 and 2 imply that $d_j''' \geq d_j'', j = 0, 1, 2, \dots .$ Therefore causal dominance implies that the column distances cannot be decreased. Also, equations (83) and (84) imply that $m''' \leq m''$, where m''' is the memory of G''' and m'' is the memory of G'' .

At this point, steps 2 and 3 are repeated until an encoder \tilde{G} is obtained such that $\text{rank}[\tilde{G}_0] = K$, $\tilde{m} \leq m$, and \tilde{G} is causally dominant to G . Then a $K \times K$ submatrix $\tilde{R}(D)$ of

$\tilde{G}(D)$ can be found such that $\det \begin{bmatrix} \tilde{R}(D) \end{bmatrix}$ has a non-zero constant term. Hence from theorem 3.1 a systematic encoder \hat{G} with $\hat{G}(D) = \tilde{R}^{-1}(D) \tilde{G}(D)$ can be found which is rationally equivalent to and causally dominant to G . These results are summarized in the following definition and theorem.

Definition 3.5 The encoder G'' is causally dominant to the encoder G'' if $(CDOS)_{G''} \supset (CDOS)_{G''}.$

Theorem 3.2 For an encoder G with a matrix of generator functions $G(D)$, if no $K \times K$ submatrix $R(D)$ of $G(D)$ exists such that $\det \begin{bmatrix} R(D) \end{bmatrix}$ has a non-zero constant term, then a systematic encoder \hat{G} can be found which is rationally equivalent to and causally dominant to G .

If the above procedure never results in an encoder \tilde{G} such that a $K \times K$ submatrix $\tilde{R}(D)$ of $\tilde{G}(D)$ can be found with $\det \begin{bmatrix} \tilde{R}_0 \end{bmatrix} \neq 0$, then it can be shown that the procedure produces a matrix with an all-zero row after at most $Km' + 1$ applications of step 3, where m' is the memory of the polynomial encoder G' . This follows because each time step 3 is applied, the memory of one of the K generators must be reduced. In this case the original encoder G has no inverse and $d_j = 0$, $j = 0, 1, 2, \dots$.

Example 3.3 Consider the $R = 2/3$ fixed binary non-systematic encoder G with $G(D) = \begin{bmatrix} 1 & 1+D & 1+D^2 \\ 1+D & 1+D+D^2 & 1 \end{bmatrix}$.

$\underline{G}_0 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and $\text{rank} \begin{bmatrix} \underline{G}_0 \end{bmatrix} = 1$. After adding row 1 to row 2 and multiplying row 2 by D^{-1} , the encoder \tilde{G} with

$\tilde{G}(D) = \begin{bmatrix} 1 & 1+D & 1+D^2 \\ 1 & D & D \end{bmatrix}$ results. Choosing
 $\tilde{R}(D) = \begin{bmatrix} 1 & 1+D \\ 1 & D \end{bmatrix}$, $\det[\tilde{R}(D)] = 1$ and $\tilde{R}^{-1}(D) = \begin{bmatrix} D & 1+D \\ 1 & 1 \end{bmatrix}$.
Hence $\hat{G}(D) = \tilde{R}^{-1}(D) \tilde{G}(D) = \begin{bmatrix} 1 & 0 & D^2 + D^3 \\ 0 & 1 & 1+D+D^2 \end{bmatrix}$ is the matrix
of generator functions for the systematic encoder \hat{G} which is
rationally equivalent to and causally dominant to G .|

It should be noted here that Forney's [6] method of producing a "canonic" non-systematic encoder does not consider the problem of preserving column distance.

The results of Section III.A can be summarized by stating that there is no loss of generality in considering only systematic encoders for algebraic feedback decoding techniques where column distance is the important parameter. However, in many cases a non-systematic polynomial encoder G' with encoding memory m' may have a causally dominant, systematic, rationally equivalent encoder \hat{G} with infinite encoding memory, i.e., $\hat{G}(D)$ may contain rational functions. Since it is often undesirable to have feedback in the encoder, \hat{G} can be converted to a systematic polynomial encoder \hat{G}' of approximately the same complexity as the encoder G' by truncating each generator function in $\hat{G}(D)$ after degree m' . Clearly the encoding memory of \hat{G}' is m' and $\hat{d}'_j = \hat{d}_j \geq d'_j$, $j = 0, 1, \dots, m'$. But the encoder \hat{G}' is no longer rationally equivalent to G' , and hence may have a lower value of d_{FREE} .

Therefore the code produced by the systematic encoder \hat{G}' may not perform as well with sequential decoding as the code produced by the non-systematic encoder G' with the same encoding memory.

B. Syndrome Formation

The parity-check matrix for all systematic encoders was given in Chapter I. In this section parity-check matrices and syndrome forming circuits will be given for all fixed non-systematic polynomial encoders G such that $\text{rank} \begin{bmatrix} G_0 \\ G_1 \end{bmatrix} = K$. These are then valid parity-check matrices and syndrome forming circuits for all rationally equivalent encoders (cf. equations (30) through (36)). That every fixed non-systematic encoder is rationally equivalent to a fixed non-systematic polynomial encoder G such that $\text{rank} \begin{bmatrix} G_0 \\ G_1 \end{bmatrix} = K$ was shown in the previous section.

The parity-check matrix for $R = \frac{1}{N}$ fixed non-systematic encoders is given by

$$\underline{H}^T(D) = \left[\begin{array}{cccc} G_1^{(2)}(D) & G_1^{(3)}(D) & \dots & G_1^{(N)}(D) \\ G_1^{(1)}(D) & 0 & \dots & 0 \\ 0 & G_1^{(1)}(D) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & G_1^{(1)}(D) \end{array} \right] . \quad (87)$$

The syndrome forming circuit for these encoders is shown in Figure 3.1.

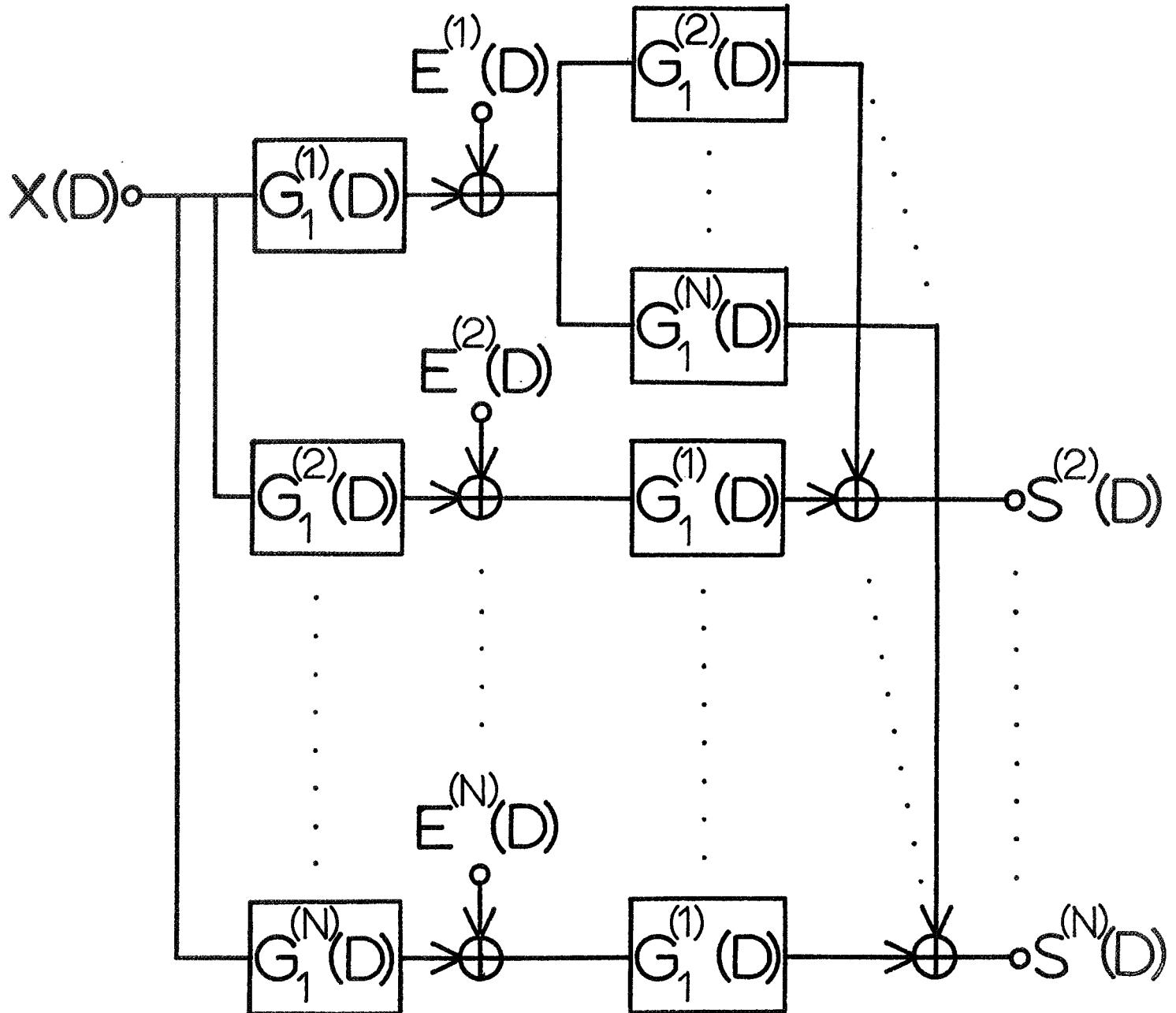


Fig. 3.1. Syndrome forming circuit for $R = \frac{1}{N}$ encoders.

However for non-systematic encoders with $K \geq 2$, $\underline{H}^T(D)$ contains products of generator functions. Consider for example an $R = 2/3$ fixed non-systematic encoder with

$$\underline{G}(D) = \begin{bmatrix} G_1^{(1)}(D) & G_1^{(2)}(D) & G_1^{(3)}(D) \\ G_2^{(1)}(D) & G_2^{(2)}(D) & G_2^{(3)}(D) \end{bmatrix} \quad (88)$$

and

$$\underline{H}^T(D) = \begin{bmatrix} H^{(1)}(D) \\ H^{(2)}(D) \\ H^{(3)}(D) \end{bmatrix} . \quad (89)$$

Since $\underline{G}(D) \underline{H}^T(D) = \underline{0}$,

$$\begin{bmatrix} G_1^{(1)}(D) & G_1^{(2)}(D) & G_1^{(3)}(D) \\ G_2^{(1)}(D) & G_2^{(2)}(D) & G_2^{(3)}(D) \end{bmatrix} \begin{bmatrix} H^{(1)}(D) \\ H^{(2)}(D) \\ H^{(3)}(D) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (90)$$

Therefore

$$G_1^{(1)}(D) H^{(1)}(D) + G_1^{(2)}(D) H^{(2)}(D) + G_1^{(3)}(D) H^{(3)}(D) = 0, \quad (91)$$

$$G_2^{(1)}(D) H^{(1)}(D) + G_2^{(2)}(D) H^{(2)}(D) + G_2^{(3)}(D) H^{(3)}(D) = 0, \quad (92)$$

and

$$H^{(1)}(D) = \frac{G_1^{(2)}(D) H^{(2)}(D) + G_1^{(3)}(D) H^{(3)}(D)}{G_1^{(1)}(D)} . \quad (93)$$

Hence

$$\begin{aligned} & \frac{G_2^{(1)}(D) G_1^{(2)}(D) H^{(2)}(D) + G_2^{(1)}(D) G_1^{(3)}(D) H^{(3)}(D)}{G_1^{(1)}(D)} \\ & + G_2^{(2)}(D) H^{(2)}(D) + G_2^{(3)}(D) H^{(3)}(D) = 0 \end{aligned} \quad (94)$$

and

$$H^{(2)}(D) = \frac{H^{(3)}(D) \left[G_2^{(3)}(D) + \frac{G_2^{(1)}(D)G_1^{(3)}(D)}{G_1^{(1)}(D)} \right]}{G_2^{(2)}(D) + \frac{G_2^{(1)}(D)G_1^{(2)}(D)}{G_1^{(1)}(D)}} . \quad (95)$$

Now choose

$$H^{(3)}(D) = G_1^{(1)}(D)G_2^{(2)}(D) + G_2^{(1)}(D)G_1^{(2)}(D) . \quad (96)$$

Then

$$H^{(2)}(D) = G_1^{(1)}(D)G_2^{(3)}(D) + G_2^{(1)}(D)G_1^{(3)}(D) . \quad (97)$$

Finally

$$H^{(1)}(D) = \frac{G_1^{(2)}(D) \left[G_1^{(1)}(D)G_2^{(3)}(D) + G_2^{(1)}(D)G_1^{(3)}(D) \right]}{G_1^{(1)}(D)} \\ + \frac{G_1^{(3)}(D) \left[G_1^{(1)}(D)G_2^{(2)}(D) + G_2^{(1)}(D)G_1^{(2)}(D) \right]}{G_1^{(1)}(D)} \quad (98)$$

$$= G_1^{(2)}(D)G_2^{(3)}(D) + G_1^{(3)}(D)G_2^{(2)}(D) . \quad (99)$$

Therefore if \bar{m} is chosen according to equation (53),

$$\bar{m} = \max [m_{11}m_{22}, m_{21}m_{12}, m_{11}m_{23}, m_{21}m_{13}, m_{12}m_{23}, m_{13}m_{22}] , \quad (100)$$

where $m_{ij} = \text{degree } [G_i^{(j)}(D)]$ for all i and j , and the decoding memory, or syndrome circuit memory, is on the order of twice the encoding memory.

The syndrome forming circuit for $R = 2/3$ fixed non-systematic encoders is shown in Figure 3.2.

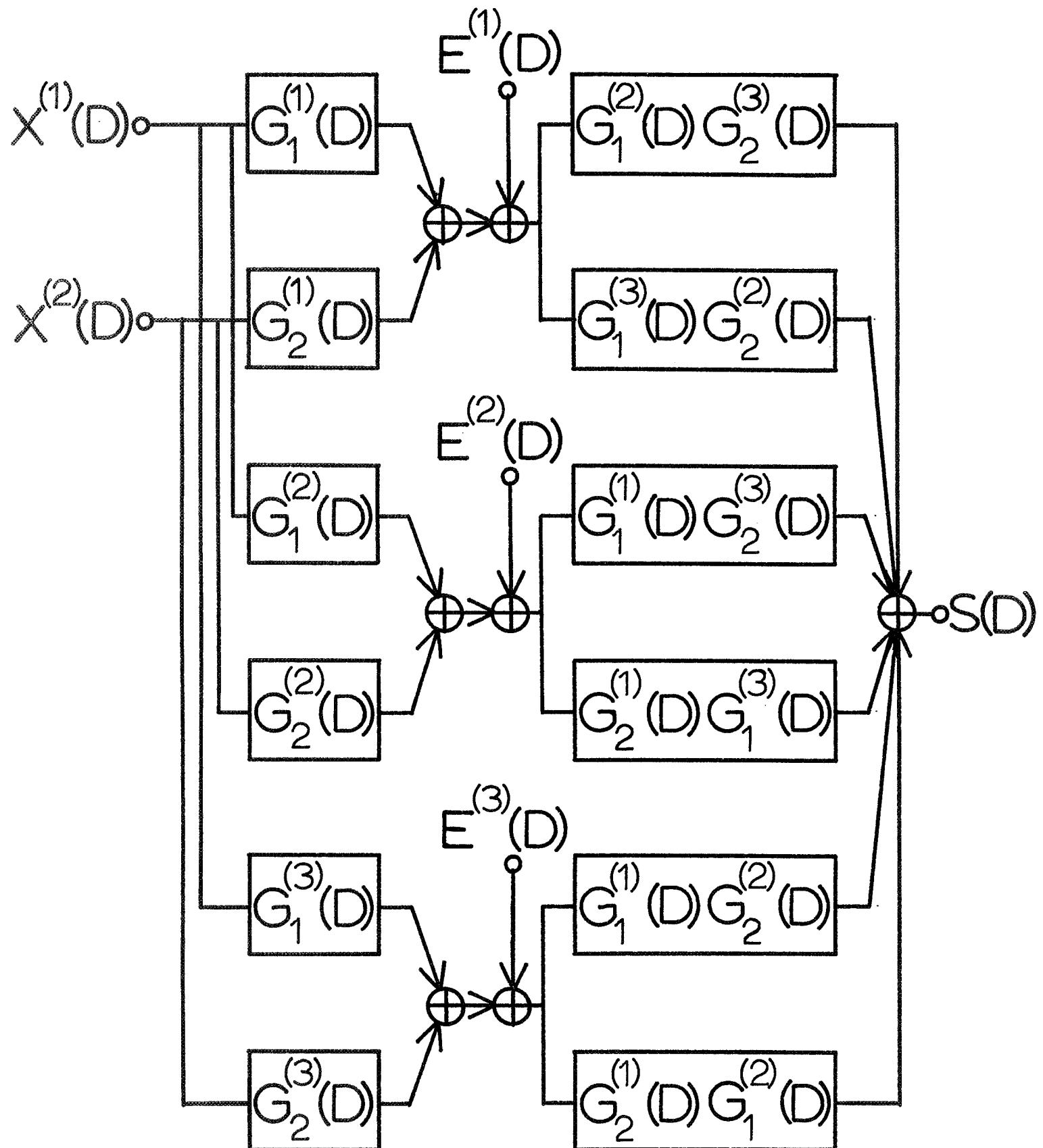


Fig. 3.2. Syndrome forming circuit for $R = \frac{2}{3}$ encoders.

Example 3.4 Consider the encoder described by

$$\underline{G}(D) = \begin{bmatrix} 1 & 1+D & 1+D+D^2 \\ D^2 & 1+D+D^3 & 1 \end{bmatrix} .$$

Then

$$\underline{H}^T(D) = \begin{bmatrix} D+D^4+D^5 \\ 1+D^2+D^3+D^4 \\ 1+D+D^2 \end{bmatrix} .$$

In this case the encoding memory $m = 3$ and, if chosen according to equation (53), the syndrome circuit memory $\bar{m} = 5$.

The syndrome equations from time u through time $u+5$ are then given by

$$\begin{bmatrix} s_u \\ s_{u+1} \\ s_{u+2} \\ s_{u+3} \\ s_{u+4} \\ s_{u+5} \end{bmatrix} = \begin{bmatrix} 110010 & 011101 & 000111 \\ 110010 & 011101 & 000111 \\ 110010 & 011101 & 000111 \\ 110010 & 011101 & 000111 \\ 110010 & 011101 & 000111 \\ 110010 & 011101 & 000111 \end{bmatrix} \begin{bmatrix} e_{u-5}^{(1)} \\ \vdots \\ e_u^{(1)} \\ \vdots \\ e_{u+5}^{(1)} \\ e_{u-5}^{(2)} \\ \vdots \\ e_u^{(2)} \\ \vdots \\ e_{u+5}^{(2)} \\ e_{u-5}^{(3)} \\ \vdots \\ e_u^{(3)} \\ \vdots \\ e_{u+5}^{(3)} \end{bmatrix} . \quad (101)$$

By generalizing from the $R = 2/3$ case, the form of the parity-check matrix for all fixed non-systematic encoders can readily be obtained. Let $\underline{R}(D)$ be the first K columns of $\underline{G}(D)$. Assume $\text{rank} \begin{bmatrix} \underline{R}_0 \\ \vdots \\ \underline{R}_K \end{bmatrix} = K$. Then denote $\det \begin{bmatrix} \underline{R}(D) \end{bmatrix}$ by Δ and the cofactor of $G_i^{(j)}(D)$ in $\underline{R}(D)$ by Δ_{ij} . Then

$$\underline{H}^T(D) = \begin{bmatrix} \sum_{i=1}^K \Delta_{i1} G_i^{(K+1)}(D) & \sum_{i=1}^K \Delta_{i1} G_i^{(K+2)}(D) & \dots & \sum_{i=1}^K \Delta_{i1} G_i^{(N)}(D) \\ \sum_{i=1}^K \Delta_{i2} G_i^{(K+1)}(D) & \sum_{i=1}^K \Delta_{i2} G_i^{(K+2)}(D) & \dots & \sum_{i=1}^K \Delta_{i2} G_i^{(N)}(D) \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^K \Delta_{iK} G_i^{(K+1)}(D) & \sum_{i=1}^K \Delta_{iK} G_i^{(K+2)}(D) & \dots & \sum_{i=1}^K \Delta_{iK} G_i^{(N)}(D) \\ \Delta & 0 & & 0 \\ 0 & \Delta & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Delta \end{bmatrix}. \quad (102)$$

It can easily be verified that equation (102) reduces to the familiar form of $\underline{H}^T(D)$ for $R = 2/3$ and $R = 1/N$ fixed non-systematic encoders and for all fixed systematic encoders.

Note that for fixed systematic encoders $\Delta = 1$. The syndrome forming circuits for $R = 2/N$ and $R = 3/N$ fixed non-systematic encoders are given in Figures 3.3 and 3.4, respectively. Equation (102) has also been implicitly derived by Forney [6].

For a fixed non-systematic encoder G such that $\text{rank} \begin{bmatrix} G_0 \\ \vdots \\ G_K \end{bmatrix} = K$, the code produced by G is exactly the same as the null space of \underline{H}^T . However, as discussed in Chapter I, if

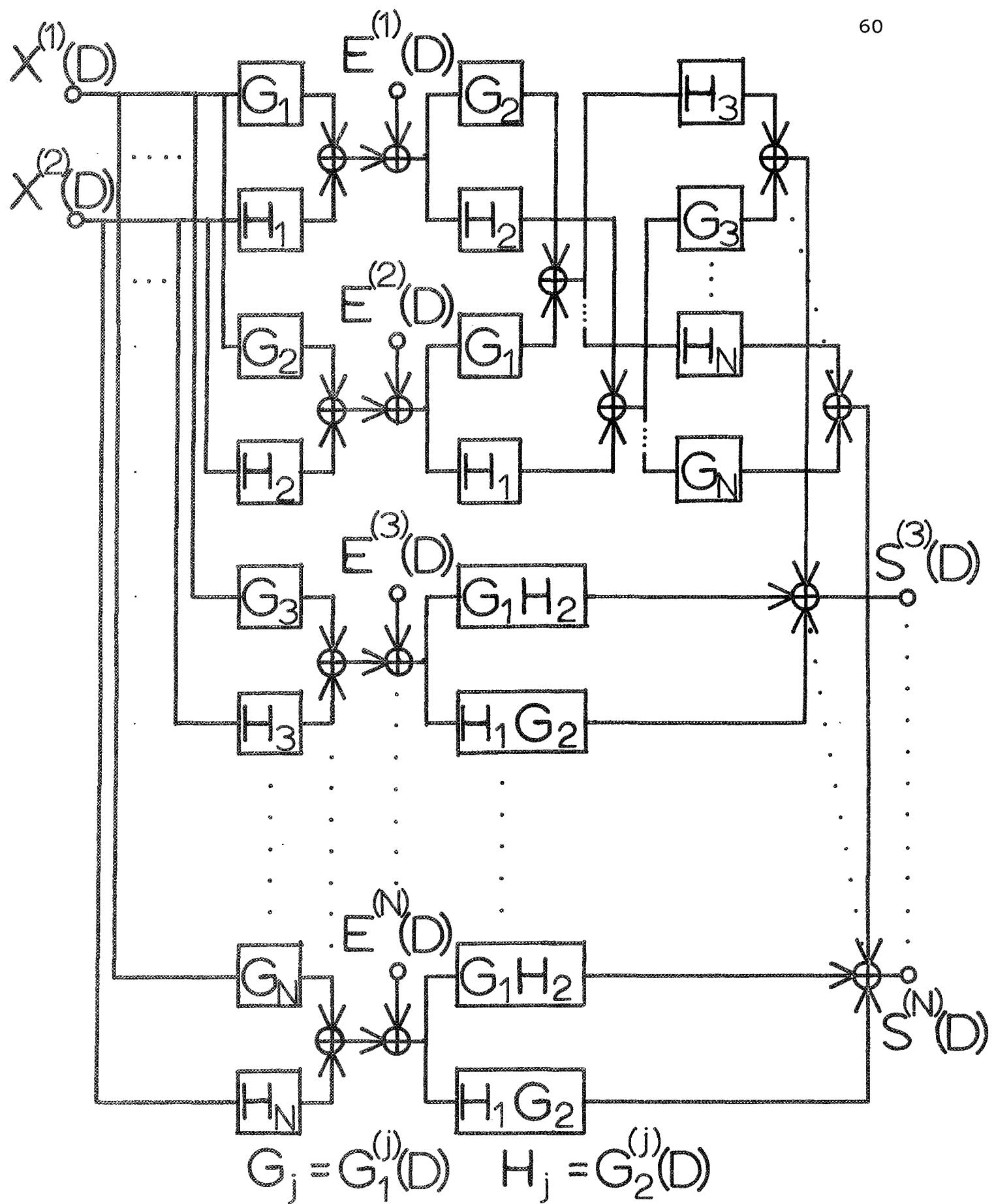


Fig. 3.3. Syndrome forming circuit for $R = \frac{2}{N}$ encoders.

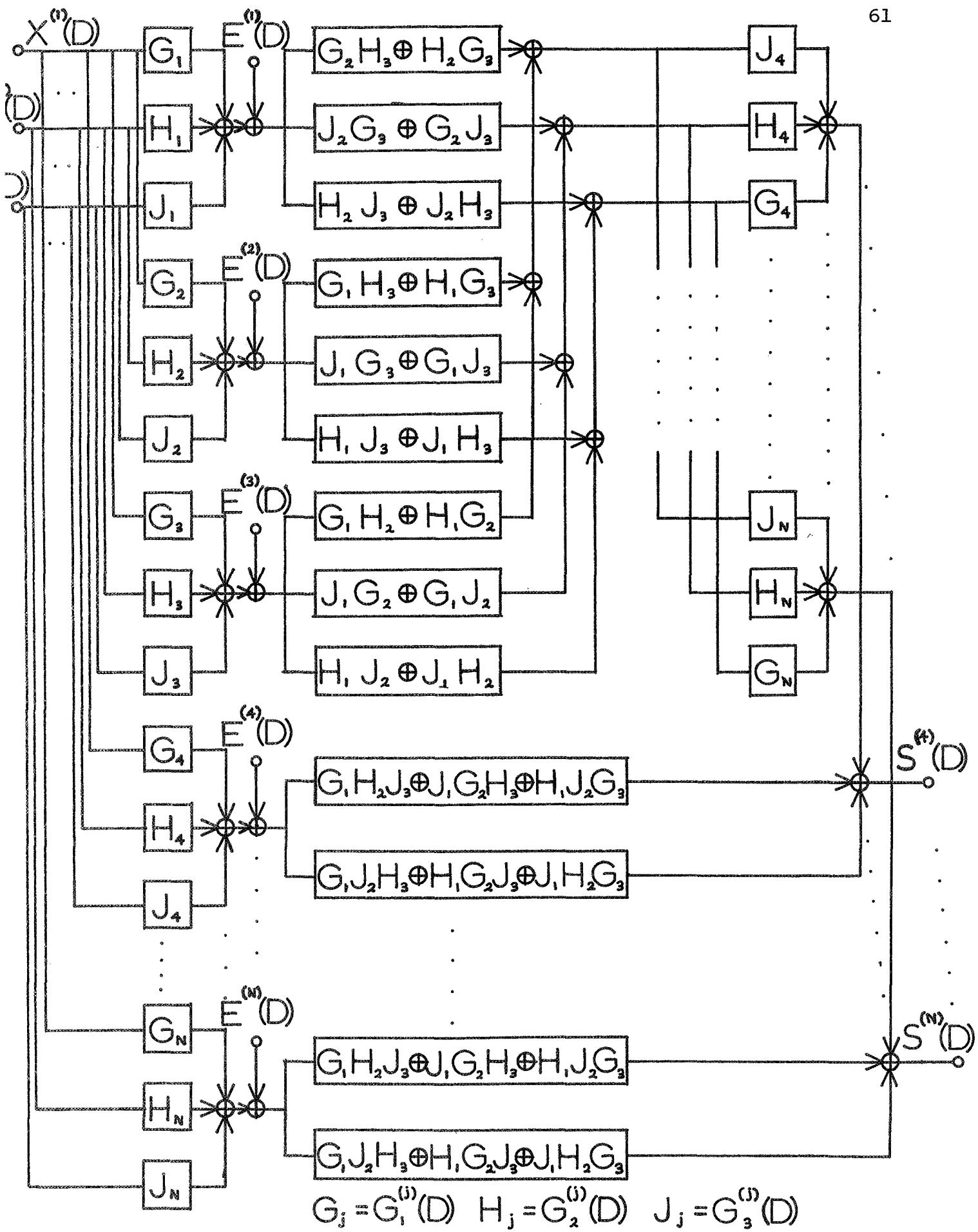


Fig. 3.4. Syndrome forming circuit for $R = \frac{3}{N}$ encoders.

rank $\begin{bmatrix} \underline{G} \\ 0 \end{bmatrix}$ < K, the null space of \underline{H}^T contains other sequences besides the set of output sequences of G. In some instances, as in Chapter VII, it is convenient to define a code with a parity-check matrix. In order to avoid ambiguity, the code defined by a parity-check matrix will be taken to be precisely the null space of \underline{H}^T .

In keeping with standard usage, throughout the remainder of this thesis many of the properties precisely defined for encoders will be referred to as properties of the code produced by a given encoder. For example, an $R = 1/N$ fixed systematic code is the code produced by an $R = 1/N$ fixed systematic encoder.

IV. Bounds on Distance

A. Introduction

A complete set of bounds on the distance properties of convolutional codes does not yet exist. However, many results in this direction have been obtained. Wozencraft and Reiffen [3] proved a Gilbert lower bound on d_{FD} for $R = \frac{1}{N}$ binary fixed codes which Massey [4] later generalized to all rates and to $GF(q)$. Robinson [16] proved an upper bound on d_{FD} for fixed codes that is asymptotically a Plotkin bound, and Massey [8] later gave a simple bound that has the same asymptotic form.

Robinson [7] also obtained a lower bound on d_{DD} for systematic fixed codes, in which d_{DD} grows only as the square root of n_{DD} . Kolor [17], for $R = 1/2$ systematic binary fixed codes only, and then Massey [8] for all rates, proved a lower bound on d_{DD} in which d_{DD} grows linearly with n_{DD} .

Wagner [18] obtained a lower bound on d_{DD} for non systematic periodic codes with $R \leq 1/2$. A new lower bound good for all rates will be given in this chapter. Wagner's results will also be extended to obtain a Gilbert lower bound on d_{FD} for a subclass of periodic codes with period $T = 2m + 1$ which does not include fixed codes as a special case. Since fixed codes are a special case of the entire class of periodic codes, the usual Gilbert lower bound on d_{FD} holds over the whole ensemble of periodic codes.

No upper bounds on d_{DD} are known except those that hold trivially since $d_{DD} \leq d_{FD}$.

Since $d_{FD} \leq d_{FREE}$, the usual Gilbert lower bound on d_{FD} is also a bound on d_{FREE} , albeit a weak one. Neumann [11] proved a much stronger lower bound on d_{FREE} for fixed non-systematic codes. In this chapter a still stronger lower bound on d_{FREE} will be given for non-systematic periodic codes and this result will be used to obtain an improved upper bound on error probability for non-systematic periodic codes used over the BSC and with a maximum likelihood decoding rule.

McEliece and Rumsey [10] obtained a Plotkin upper bound on d_{FREE} for $R = \frac{1}{N}$ systematic fixed codes. The extension of this result to all rates, to non-systematic fixed codes, and to periodic codes will be given in this chapter.

Finally a Gilbert lower bound on d_{FD} for an important subclass of $R = 1/2$ non-systematic fixed codes will be obtained. This bound has application in the chapter on code construction.

B. Bounds on d_{FD}

1. Lower Bounds

A lower bound on distance guarantees that at least one code can be found with distance greater than or equal to the lower bound. Wozencraft and Reiffen [3] and Massey [4] have shown that there exists at least one binary fixed code such that

$$\lim_{m \rightarrow \infty} \frac{d_{FD}}{n_{FD}} \geq H^{-1}(1 - R) , \quad (103)$$

where $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function. Equation (103) is called a Gilbert lower bound because it is asymptotically the same as Gilbert's lower bound on the minimum distance of a block code [19].

2. Upper Bounds

An upper bound on distance guarantees that no code can be found with distance greater than the upper bound. Robinson [16] and Massey [8] have shown that for all binary fixed codes

$$\lim_{m \rightarrow \infty} \frac{d_{FD}}{n_{FD}} \leq \frac{1}{2} (1 - R). \quad (104)$$

Equation (104) is called a Plotkin upper bound because it is asymptotically the same as Plotkin's upper bound on the minimum distance of a block code [20]. An upper bound which is asymptotically the same as Hamming's [21] upper bound on the minimum distance of a block code is called a Hamming upper bound and an upper bound which is asymptotically the same as Elias's [22] upper bound on the minimum distance of a block code is called an Elias upper bound. However no Hamming or Elias upper bounds are yet known on d_{FD} for either periodic or fixed codes.

C. Bounds on d_{DD}

1. Fixed Codes

Generalizing the work of Kolor [17], Massey [8] has shown that there exists at least one binary systematic code such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq H^{-1}\left(\frac{1}{10}, \frac{1-R}{1+R}\right). \quad (105)$$

Note that this bound guarantees a linear increase of d_{DD} with n_{DD} . In Robinson's [7] earlier bound, d_{DD} is guaranteed to increase only as the square root of n_{DD} .

The Plotkin upper bound on d_{FD} of equation (104) also holds for d_{DD} since $d_{DD} \leq d_{FD}$, but no other upper bounds on d_{DD} are known for fixed codes.

2. Periodic Codes

For periodic codes, Wagner [18] has shown that there exists at least one code such that

$$\lim_{m \rightarrow \infty} \frac{d_w}{n_w} \geq H^{-1}(1 - 2R) , \quad (106)$$

where

$$d_w = \min_{\bar{m} \leq u < \bar{m}+T} \min_{\underline{x}_u \neq \underline{x}_{u+\bar{m}}} d_H([\underline{x} G]_{u, u+\bar{m}}, [\underline{x}' G]_{u, u+\bar{m}}) = \min_{\bar{m} \leq u < \bar{m}+T} \min_{\underline{x}_u \neq \underline{x}_{u+\bar{m}}} d_H([\underline{y}]_{u, u+\bar{m}}, [\underline{y}']_{u, u+\bar{m}}) \text{ and } n_w = n_{FD}$$

are just different definitions of definite decoding minimum distance and definite decoding constraint length than those given in Chapter II. Note that this bound is only good for $R \leq 1/2$. A bound which is good for all rates can easily be derived using Robinson's and Massey's more natural definitions of the definite decoding parameters. First consider only canonic systematic codes with $\bar{m} = m$.

For a canonic systematic code with period $T = m + 1$, the equations for the parity sequence from time $2m$ through time $3m$ can be written as

$$p_{2m} = \underline{x}_{2m} \underline{Q}_0^{(m-1)} + \underline{x}_{2m-1} \underline{Q}_1^{(m-1)} + \dots + \underline{x}_m \underline{Q}_m^{(m-1)}$$

$$p_{2m+1} = x_{2m+1} \underline{Q}_0(m) + x_{2m} \underline{Q}_1(m) + \dots + x_{m+1} \underline{Q}_m(m) \quad (107)$$

⋮

⋮

$$p_{3m} = x_{3m} \underline{Q}_0(m-2) + x_{3m-1} \underline{Q}_1(m-2) + \dots + x_{2m} \underline{Q}_m(m-2),$$

where each $\underline{Q}_i(u)$, $0 \leq i, u \leq m$, is defined as in equations (16) and (17).

Theorem 4.1* There exists at least one systematic periodic time-varying convolutional code such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq H^{-1} \left(\frac{1 - R}{1 + R} \right).$$

In particular, there exists at least one periodic code with period $T = m + 1$ such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq H^{-1} \left(\frac{1 - R}{1 + R} \right).$$

Proof For a given definite decoding codeword $[x_m, \dots, x_{2m-1}, x_{2m}, \dots, x_{3m}] = [x_m, \dots, x_{2m-1}, x_{2m}, p_{2m}, \dots, x_{3m}, p_{3m}]$, there are $(m+1)^2 K(N-K)$ unknowns in equation (107). For $x_{2m} \neq 0$, equations (107) are linearly independent. Therefore since there are $(m+1)(N-K)$ equations, each codeword with $x_{2m} \neq 0$ has $\gamma = \frac{2^{(m+1)^2 K(N-K)}}{2^{(m+1)(N-K)}}$ solutions. Finally, since $T = m + 1$, each codeword with some $x_u \neq 0$, $2m \leq u < 2m + T$, belongs to at most $(m+1)\gamma$ different codes.

The number of codewords with Hamming weight less than

or equal to d is $\sum_{i=0}^d \binom{n_{DD}}{i} \leq 2^{n_{DD}} H\left(\frac{d}{n_{DD}}\right)$ when $\frac{d}{n_{DD}} \leq 1/2$
[3].

* This result was obtained independently by Morrissey [23].

Hence the number of codewords with $\underline{x}_{2m} \neq 0$ and weight less than or equal to d is less than $2^{n_{DD}} H\left(\frac{d}{n_{DD}}\right)$. Therefore if $(m+1)\gamma 2^{n_{DD}} H\left(\frac{d}{n_{DD}}\right)$ is less than the total number of codes, there exists at least one code with $d_{DD} > d$. Equivalently, if d_{DD} is the smallest integer such that $(m+1)\gamma 2^{n_{DD}} H\left(\frac{d_{DD}}{n_{DD}}\right) \geq 2^{(m+1)^2 K(N-K)}$, then there exists at least one code with definite decoding minimum distance d_{DD} . But

$$\begin{aligned}
 & (m+1)\gamma 2^{[mK + (m+1)N]} H\left(\frac{d_{DD}}{n_{DD}}\right) \geq 2^{(m+1)^2 K(N-K)} \longrightarrow \\
 & \log_2 (m+1) + (m+1)^2 K(N-K) - (m+1)(N-K) + [mK + \\
 & (m+1)N] H\left(\frac{d_{DD}}{n_{DD}}\right) \geq (m+1)^2 K(N-K) \longrightarrow \\
 & \log_2 (m+1) - (m+1)(N-K) + [mK + (m+1)N] H\left(\frac{d_{DD}}{n_{DD}}\right) \geq 0 \longrightarrow \\
 & [mK + (m+1)N] \left[H\left(\frac{d_{DD}}{n_{DD}}\right) + \frac{\log_2 (m+1)}{mK+(m+1)N} - \frac{(m+1)(N-K)}{mK+(m+1)N} \right] \geq 0 \longrightarrow \\
 & H\left(\frac{d_{DD}}{n_{DD}}\right) \geq \frac{(N-K)(m+1)}{mK+(m+1)N} - \frac{\log_2 (m+1)}{mK+(m+1)N} \longrightarrow \\
 & H\left(\frac{d_{DD}}{n_{DD}}\right) \geq \frac{1-R}{1+R} \quad \text{as } m \rightarrow \infty \longrightarrow \\
 & \lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq H^{-1}\left(\frac{1-R}{1+R}\right). |
 \end{aligned}$$

This bound is plotted and compared with the usual Gilbert lower bound on d_{FD} in Figure 4.1.

Massey [8] has conjectured that his bound on d_{DD} for systematic fixed codes should be the same as the bound in theorem 4.1. He claims that the factor of $\frac{1}{10}$ in equation (105) should be eliminated by tighter arguments.

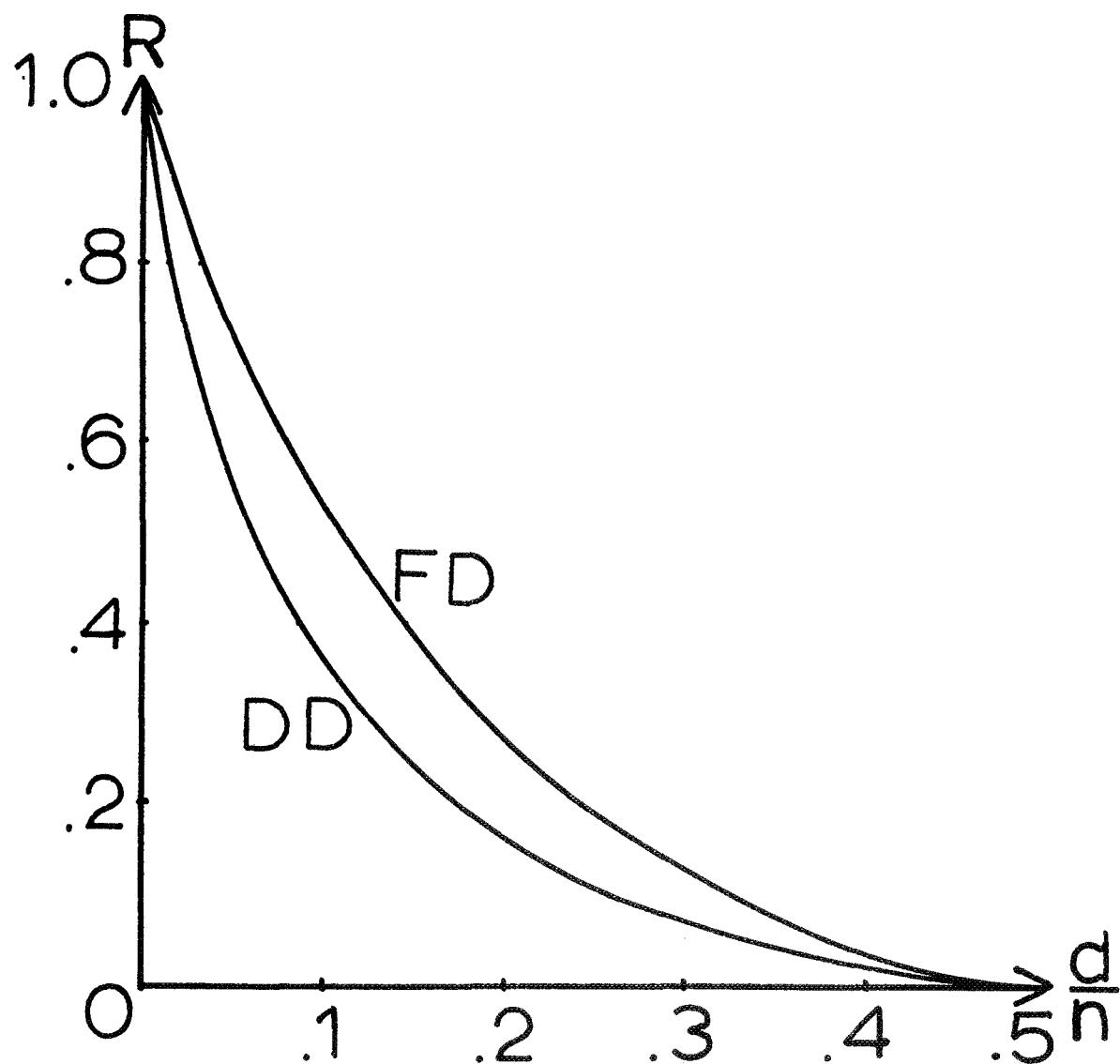


Fig. 4.1. Comparison of feedback decoding and definite decoding bounds.

For non-systematic codes with decoding memory $\bar{m} = m$ and period $T = 2m + 1$, the equations for the transmitted sequence from time m through time $3m$ can be written as

$$\begin{aligned} \underline{y}_m &= \underline{x}_m \underline{G}_0(m) + \underline{x}_{m-1} \underline{G}_1(m) + \dots + \underline{x}_0 \underline{G}_m(m) \\ &\quad \cdot \\ &\quad \cdot \\ &\quad \cdot \\ \underline{y}_{2m} &= \underline{x}_{2m} \underline{G}_0(2m) + \underline{x}_{2m-1} \underline{G}_1(2m) + \dots + \underline{x}_m \underline{G}_m(2m) \\ &\quad \cdot \\ &\quad \cdot \\ &\quad \cdot \\ \underline{y}_{3m} &= \underline{x}_{3m} \underline{G}_0(m-1) + \underline{x}_{3m-1} \underline{G}_1(m-1) + \dots + \underline{x}_{2m} \underline{G}_m(m-1) . \end{aligned} \quad (108)$$

Theorem 4.2 There exists at least one non-systematic periodic time-varying convolutional code such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq \min \left[H^{-1} \left(1 - \frac{3}{2}R \right), \frac{1}{2} H^{-1}(1-R) \right] .$$

In particular, there exists at least one periodic code with period $T = 2m + 1$ such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq \min \left[H^{-1} \left(1 - \frac{3}{2}R \right), \frac{1}{2} H^{-1}(1-R) \right] .$$

Proof For a given definite decoding codeword $[\underline{y}_m, \dots, \underline{y}_{2m}, \dots, \underline{y}_{3m}]$, there are $(m+1)^2 N K$ unknowns and $(2m+1)N$ equations in equations (108). But $\underline{x}_{2m} \neq \underline{0}$ guarantees only that the last $(m+1)N$ equations are linearly independent.

Information blocks \underline{x}_0 through \underline{x}_{2m-1} could be all-zero.

Assume \underline{x}_j is the first non-zero information block, $j = 0, 1, \dots, 2m$. If $j \geq m$, then the usual Gilbert lower bound on d_{FD} holds for the definite decoding minimum distance, i.e.,

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{FD}} \geq H^{-1}(1 - R) \text{ or } \lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq 1/2 H^{-1}(1 - R) \text{ since}$$

$\bar{m} = m$ implies that $n_{DD} = (2m + 1)N \approx 2(m + 1)N = 2n_{FD}$ for large m . However, if $j < m$, all codewords produced by an information sequence containing a span of m consecutive all-zero information blocks inclusive between block j and block $(2m - 1)$ have Hamming weight at least as great as the minimum weight codeword with $\underline{x}_0 = \dots = \underline{x}_{2m-1} = \underline{0}$, i.e., at least as great as d_{FD} . Hence all such codewords need not be considered since they cannot possibly be the minimum weight codeword with $\underline{x}_{2m} \neq \underline{0}$. Therefore all of equations (108) are independent, and for each codeword with $\underline{x}_{2m} \neq \underline{0}$ there are

$$\gamma = \frac{2^{NK(m+1)(2m+1)}}{2^{N(2m+1)}}$$

solutions to equations (108).

Finally, since $T = 2m + 1$ and the information digits can be chosen in any of $\frac{2^K - 1}{2^K} 2^{(3m+1)K}$ different ways, each codeword with some $\underline{x}_u \neq \underline{0}$, $2m \leq u < 2m+T$, belongs to at most $(2m + 1) \frac{2^{K-1}}{2^K} 2^{(3m+1)K} \gamma < (2m + 1) 2^{(3m+1)K} \gamma$ different codes.

Therefore, proceeding as in the proof of theorem 4.1, if d_{DD} is the smallest integer such that $(2m + 1) 2^{(3m+1)K} \gamma 2^{n_{DD}} H\left(\frac{d_{DD}}{n_{DD}}\right) \geq 2^{NK(m+1)(2m+1)}$, then there exists at least one code with definite decoding minimum distance d_{DD} . But $(2m + 1) 2^{(3m+1)K} \gamma 2^{(2m+1)N} H\left(\frac{d_{DD}}{n_{DD}}\right) \geq 2^{NK(m+1)(2m+1)} \longrightarrow$ $\log_2 (2m+1) + (3m+1)K - N(2m+1) + (2m+1)N H\left(\frac{d_{DD}}{n_{DD}}\right) \geq 0 \longrightarrow$

$$(2m+1)N \left[H \left(\frac{d_{DD}}{n_{DD}} \right) - 1 + \frac{(3m+1)K}{(2m+1)N} + \frac{\log_2(2m+1)}{(2m+1)N} \right] \geq 0 \longrightarrow$$

$$H \left(\frac{d_{DD}}{n_{DD}} \right) - 1 + 3/2 R \geq 0 \text{ as } m \rightarrow \infty \longrightarrow$$

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq H^{-1}(1 - 3/2 R) .$$

Therefore

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq \min \left[H^{-1}(1 - 3/2 R), 1/2 H^{-1}(1 - R) \right] .$$

Note that this bound holds only for $R \leq 2/3$.

Theorem 4.2 can be extended to all rates by redefining d_{DD} and n_{DD} . Assume $n_{DD} = (3m+1)N$ and $T = 3m+1$. Then $d_{DD} = \min_{3m \leq u < 3m+T} \min_{\substack{x_u \neq 0}} w_H([y]_{u-2m, u+m})$, and after a slight modification of the proof of theorem 4.2, the following corollary results.

Corollary 4.1 There exists at least one non-systematic periodic time-varying convolutional code such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq \min \left[H^{-1}(1 - 4/3 R), 1/3 H^{-1}(1 - R) \right] .$$

In general, let $n_{DD} = (\lambda m + 1)N$ and $T = \lambda m + 1$, λ a positive integer. Then $d_{DD} = \min_{\lambda m \leq u < \lambda m + T} \min_{\substack{x_u \neq 0}} w_H([y]_{u+m-\lambda m, u+m})$, and there exists at least one non-systematic periodic time-varying convolutional code such that $\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq \min \left[H^{-1}(1 - \frac{\lambda+1}{\lambda} R), \frac{1}{\lambda} H^{-1}(1-R) \right]$. Note that this reduces to Wagner's bound when $\lambda = 1$, to theorem 4.2 when $\lambda = 2$, and to corollary 4.1 when $\lambda = 3$.

Note also that the above bound holds for any $T \geq \lambda m + 1$ such that T is some algebraic function of m , i.e., such that T grows less than exponentially with m , since $\frac{\log_2 T}{N(\lambda m + 1)}$ still approaches 0 as m approaches ∞ . It is well known [24] that bounds which use a fraction of codes argument in the proof, such as the above bounds, hold for almost all codes. Hence letting λ_R be the value of λ which maximizes the above bound for a given R , the following corollary results.

Corollary 4.2 For almost all non-systematic periodic time-varying convolutional codes with period $T \geq \lambda_R^{m+1}$ such that T grows less than exponentially with m ,

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq \min \left[H^{-1} \left(1 - \frac{\lambda_R + 1}{\lambda_R} R \right), \frac{1}{\lambda_R} H^{-1} \left(1 - R \right) \right].$$

Corollary 4.2 guarantees a linear growth of d_{DD} with n_{DD} for almost all non-systematic periodic codes, even though the bound is very weak for high rates. The two functions which comprise the bound of theorem 4.2 are plotted together with the usual Gilbert lower bound on d_{FD} in Figure 4.2.

Wagner also proved that the result of equation (106) holds for an easily instrumented class of systematic codes suggested by Massey with period $T = \frac{3m+1}{2}$ if m is odd or $3m + 1$ if m is even. The $R = 1/2$ encoder in this class is shown in Figure 4.3. Once each second the top shift register shifts once while the bottom one shifts twice. It can also be shown that there exists at least one of these codes such that

$$\lim_{m \rightarrow \infty} \frac{d_{DD}}{n_{DD}} \geq H^{-1} \left(\frac{1 - R}{1 + R} \right).$$

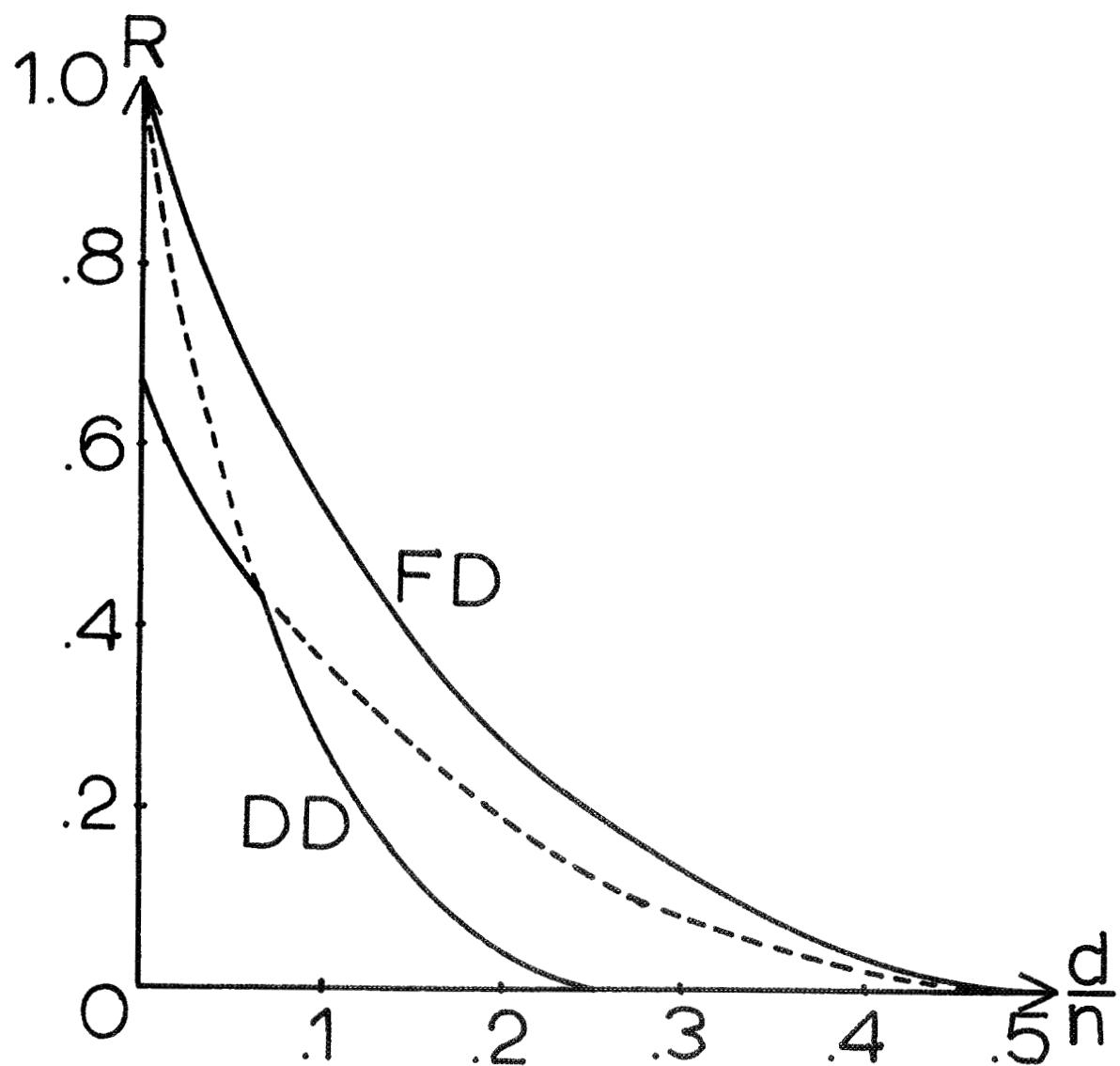


Fig. 4.2. A lower bound on d_{DD} for non-systematic periodic codes.

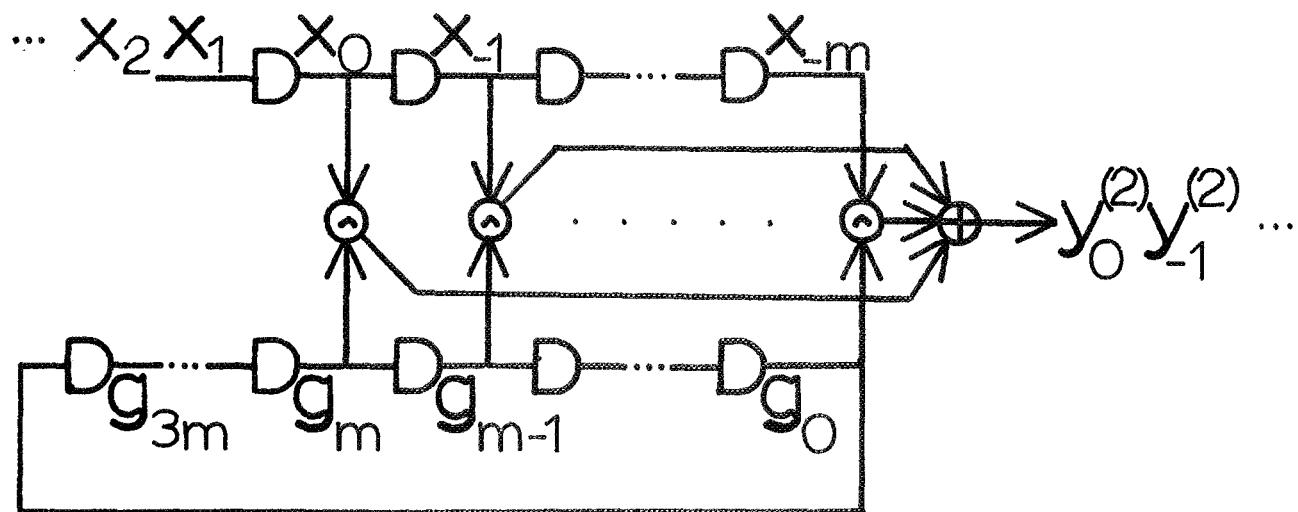


Fig. 4.3. An $R = \frac{1}{2}$ periodic encoder with $T = \frac{3m+1}{2}$ if m is odd or $T = 3m+1$ if m is even.

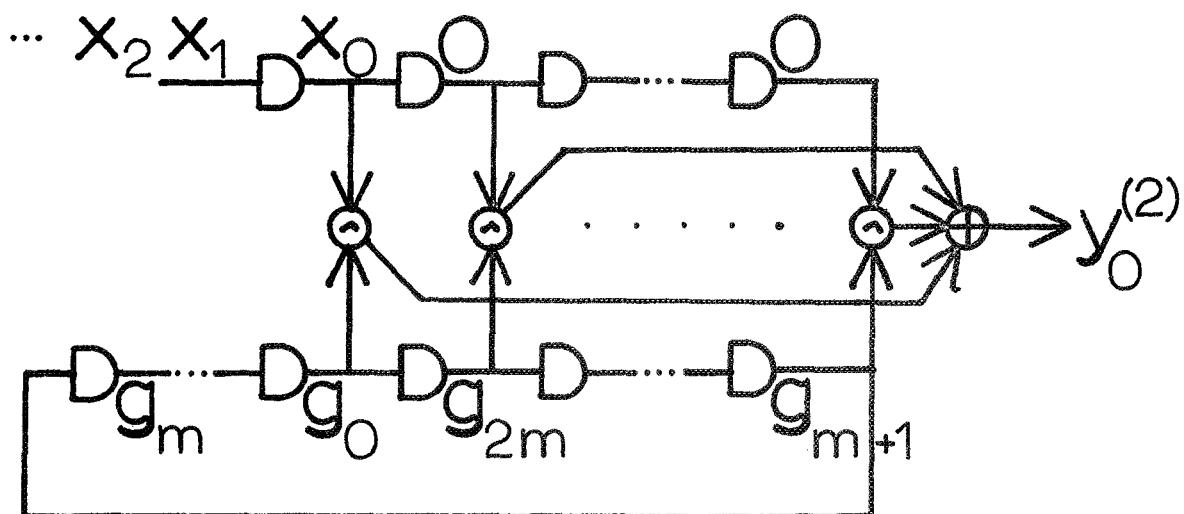


Fig. 4.4. An $R = \frac{1}{2}$ periodic encoder with $T = 2m+1$.

Now a similar class of systematic codes will be presented for which a Gilbert lower bound on d_{FD} is proved. These codes do not include fixed codes as a special case and hence such a bound is interesting. These codes have period $T = 2m + 1$ and the $R = 1/2$ encoding circuit is shown in Figure 4.4. Once each second the top shift register shifts once while the bottom one shifts twice. This leads to the following equations for the parity vectors:

$$(0) \quad p_0 = \underline{x}_0 \underline{\Omega}_0$$

$$(1) \quad p_1 = \underline{x}_1 \underline{\Omega}_2 + \underline{x}_0 \underline{\Omega}_1$$

$$(2) \quad p_2 = \underline{x}_2 \underline{\Omega}_4 + \underline{x}_1 \underline{\Omega}_3 + \underline{x}_0 \underline{\Omega}_2$$

 \vdots
 \vdots
 \vdots

$$(m-1) \quad p_{m-1} = \underline{x}_{m-1} \underline{\Omega}_{2m-2} + \underline{x}_{m-2} \underline{\Omega}_{2m-3} + \dots + \underline{x}_0 \underline{\Omega}_{m-1} \quad (109)$$

$$(m) \quad p_m = \underline{x}_m \underline{\Omega}_{2m} + \underline{x}_{m-1} \underline{\Omega}_{2m-1} + \dots + \underline{x}_0 \underline{\Omega}_m$$

$$(m+1) \quad p_{m+1} = \underline{x}_{m+1} \underline{\Omega}_1 + \underline{x}_m \underline{\Omega}_0 + \underline{x}_{m-1} \underline{\Omega}_{2m} + \dots + \underline{x}_1 \underline{\Omega}_{m+2}$$

 \vdots
 \vdots

$$(2m) \quad p_{2m} = \underline{x}_{2m} \underline{\Omega}_{2m-1} + \underline{x}_{2m-1} \underline{\Omega}_{2m-2} + \dots + \underline{x}_m \underline{\Omega}_{m-1},$$

where each $\underline{\Omega}_i$, $0 \leq i \leq 2m$, is a $K \times (N - K)$ matrix of elements from $GF(2)$.

Lemma 4.1 For each fixed sequence $[y_0, y_1, \dots, y_m] = [\underline{x}_0, p_0, \underline{x}_1, p_1, \dots, \underline{x}_m, p_m]$ with $\underline{x}_0 \neq \underline{0}$, there are

$$\gamma = \frac{2^{(2m+1)K(N-K)}}{2^{(m+1)(N-K)}}$$

solutions to the first $m+1$ of equations (109).

Proof For each codeword $[y]_m$ with $\underline{x}_0 \neq \underline{0}$, the m^{th} equation fixes one row of $\underline{\Omega}_m$. The remaining $K - 1$ rows can be

chosen arbitrarily, along with all the rows of \underline{Q}_{m+1} , \underline{Q}_{m+2} , ..., \underline{Q}_{2m} . In the $(m - 1)^{st}$ equation, then, only \underline{Q}_{m-1} has not been specified. The $(m - 1)^{st}$ equation fixes one row of \underline{Q}_{m-1} and the others can be chosen arbitrarily. Similarly in each of the remaining $m - 1$ equations, only one row of one matrix need be fixed. Therefore a total of $(m+1)(N-K)$ digits are fixed by the first $m + 1$ of equations (109). Since there are $\frac{2^{(2m+1)K(N-K)}}{(2m+1)K(N-K)}$ unknowns, there are $\frac{2^{(2m+1)K(N-K)}}{2^{(m+1)(N-K)}}$ solutions. |

Theorem 4.3 There exists at least one systematic code of the type shown in Figure 4.4 and described by equations (109) such that

$$\lim_{m \rightarrow \infty} \frac{d_{FD}}{n_{FD}} \geq H^{-1}(1 - R) .$$

Proof Since $T = 2m + 1$, each codeword $[y]_{u, u+m}$ with some $y_u \neq 0$, $0 \leq u < 2m + 1$, can belong to at most $(2m + 1) \gamma$

different codes. Therefore if d_{FD} is the smallest integer such that $(2m + 1) \gamma 2^{n_{FD}H} \left(\frac{d_{FD}}{n_{FD}}\right) \geq 2^{(2m+1)K(N-K)}$, then

there exists at least one code with feedback decoding minimum distance d_{FD} . But

$$(2m + 1) 2^{(2m+1)K(N-K) - (m+1)(N-K) + (m+1)NH} \left(\frac{d_{FD}}{n_{FD}}\right) \geq 2^{(2m+1)K(N-K)} \rightarrow$$

$$\log_2 (2m + 1) + (m + 1)NH \left(\frac{d_{FD}}{n_{FD}}\right) - (m + 1)(N - K) \geq 0 \rightarrow$$

$$(m + 1)N \left[H \left(\frac{d_{FD}}{n_{FD}}\right) - 1 + \frac{K(m+1)}{N(m+1)} + \frac{\log_2 (2m+1)}{N(m+1)} \right] \geq 0 \rightarrow$$

$$H \left(\frac{d_{FD}}{n_{FD}}\right) \geq 1 - R \quad \text{as } m \rightarrow \infty \rightarrow$$

$$\lim_{m \rightarrow \infty} \frac{d_{FD}}{n_{FD}} \geq H^{-1}(1 - R) . |$$

Theorem 4.3 guarantees that simply instrumented codes of the type shown in Figure 4.4 and described by equations (109) can be constructed with large d_{FD} .

D. Bounds on d_{FREE}

1. Lower Bounds

a. Fixed Codes

Clearly all lower bounds on d_{FD} are also lower bounds on d_{FREE} since $d_{FD} \leq d_{FREE}$. The only other lower bound on d_{FREE} was given recently by Neumann [11] for non-systematic codes only. His result states that there exists at least one binary non-systematic fixed code such that

$$\frac{d_{FREE}}{n_A} \geq \begin{cases} 2H^{-1}(1-R) & \text{for } R \geq 0.37 \\ \frac{2R(1-2^{2R-1})}{H(1-2^{2R-1})+2R-1} & \text{for } R \leq 0.37. \end{cases} \quad (110)$$

b. Periodic Codes

A stronger bound than (110) can be obtained for binary non-systematic periodic codes. First a bound similar to property F4 must be proved for non-systematic periodic codes.

Lemma 4.2 $d_{FREE} = r^{\left[\frac{N(m+1)-1}{m} \right]} \left[\frac{(\lambda_R + 1)m}{\lambda_R^{m+1}} \right]$ for almost all non-systematic periodic codes with period $T \geq \lambda_R^{m+1}$ such that T grows less than exponentially with m , where λ_R is defined as in corollary 4.2.

Proof Corollary 4.2 guarantees a linear growth of d_{DD} with n_{DD} and hence $d_{DD} > 0$. This implies that $d_{FD} > 0$ and for all information sequences \underline{x} with $x_0 \neq 0$, the first $m+1$ blocks of transmitted digits must contain at least one 1. Assume n_{DD} and d_{DD} are defined as in corollary 4.2. As in the proof of

property F4, no information sequences with m or more consecutive all-zero blocks need be considered. Hence $d_{DD} > 0$ implies that the next $(\lambda_R + 1)m$ blocks of transmitted digits must also contain at least one 1. Therefore since $d_{FREE} \leq N(m+1)$ and all the possible minimum free weight codewords must have weight at least $N(m+1)$ after $m+1 + [N(m+1)-1][(\lambda_R + 1)m]$ transmitted blocks, $d_{FREE} = r [N(m+1)-1] [(\lambda_R + 1)m] + 1$.

Let $\ell_{max} = [N(m+1)-1] [(\lambda_R + 1)m] + 1$, the bound on the length (in blocks of K digits each) of information sequence needed to produce the minimum free weight codeword. Consider the ensemble of non-systematic periodic time-varying codes with $T = \ell_{max} + m$. Clearly T is only an algebraic function of m , i.e., T grows less than exponentially with m . Let S_ℓ be the set of all information sequences of length ℓ such that $x_u \neq 0, x_{u+\ell-1} \neq 0, x_0 = x_1 = \dots = x_{u-1} = x_{u+\ell} = x_{u+\ell+1} = \dots = 0$, for some u , $0 \leq u < T$, and which contain no string of m or more all-zero blocks inclusive between block u and block $u+\ell-1$. Then let $F(\ell, d)$ be the fraction of codes with a codeword of weight d or less produced by an information sequence from the set S_ℓ .

Lemma 4.3

$$F(\ell, d) \leq \frac{T(2^{K-1}) \sum_{j=0}^d \binom{N(m+1)}{j}}{2^{N(m+1)}} .$$

Proof For a particular information sequence of length ℓ belonging to S_ℓ , the number of different ways of choosing a low weight row of G must be specified. Clearly, there are

$$\sum_{j=0}^d \binom{N(m+1)}{j} \text{ ways of choosing a low weight } N(m+1) \text{-tuple.}$$

Once one row of \underline{G} has been specified as having low weight, the digits of the remaining $[K-1 + (T-1)K]$ distinct rows can be chosen arbitrarily. Hence a low weight codeword produced by an information sequence from S_1 can appear in at most

$$2^{[K-1+(T-1)K]} \cdot 2^{N(m+1)} \sum_{j=0}^d \binom{N(m+1)}{j} = 2^{NKT(m+1)-N(m+1)} \sum_{j=0}^d \binom{N(m+1)}{j}$$

codes. Since there are $T(2^K - 1)$ ways of choosing such an information sequence, $F(1, d)$ is at most

$$\frac{T(2^K - 1) \cdot 2^{NKT(m+1)-N(m+1)} \sum_{j=0}^d \binom{N(m+1)}{j}}{\sum_{j=0}^d \binom{N(m+1)}{j}} = 2^{N(m+1)}. |$$

Lemma 4.4

$$F(\ell, d) \leq \frac{T(2^K - 1) \cdot 2^{\sum_{j=0}^d \binom{N(m+\ell)}{j}}}{2^{(N-K)\ell} \cdot 2^{2K+Nm}} \quad \text{for } \ell = 2, 3, \dots, \ell_{\max}.$$

Proof For a particular information sequence of length ℓ belonging to S_ℓ , the transmitted codeword has a length of $m+\ell$ blocks.

Hence there are $\sum_{j=0}^d \binom{N(m+\ell)}{j}$ low weight sequences that are

possible candidates for low weight codewords. The encoding equations for $u=0$ can be written as follows:

$$\underline{y}_0 = \underline{x}_0 \underline{G}_0(0)$$

$$\underline{y}_1 = \underline{x}_1 \underline{G}_0(1) + \underline{x}_0 \underline{G}_1(1)$$

$$\underline{y}_2 = \underline{x}_2 \underline{G}_0(2) + \underline{x}_1 \underline{G}_1(2) + \underline{x}_0 \underline{G}_2(2)$$

•
•
•

$$\begin{aligned}
 y_m &= x_m G_0(m) + x_{m-1} G_1(m) + \dots + x_0 G_m(m) \\
 &\vdots \\
 &\vdots \\
 y_{\ell-1} &= x_{\ell-1} G_0(\ell-1) + x_{\ell-2} G_1(\ell-1) + \dots + x_{\ell-m-1} G_m(\ell-1) \\
 &\vdots \\
 &\vdots \\
 y_{m+\ell-1} &= x_{\ell-1} G_m(m+\ell-1)
 \end{aligned} \tag{111}$$

Since no information sequences with m or more consecutive all-zero blocks are being considered, and $T \geq m + \ell$, equations (111), when put into matrix form, have rank equal to the number of equations. Therefore, given a particular information sequence from S_ℓ and letting $[y]_{m+\ell-1}$ be a particular low weight

sequence, there are $\frac{2^{TNK(m+1)}}{2^{N(m+\ell)}}$ solutions to equations (111).

Since there are at most $2^{K(\ell-2)} (2^K - 1)^2$ different information sequences in S_ℓ ,

$$\begin{aligned}
 F(\ell, d) &\leq \frac{T 2^{K(\ell-2)} (2^K - 1)^2 2^{TNK(m+1)} \sum_{j=0}^d \binom{N(m+\ell)}{j}}{2^{N(m+\ell)} 2^{TNK(m+1)}} \\
 &\leq \frac{T (2^K - 1)^2 \sum_{j=0}^d \binom{N(m+\ell)}{j}}{2^{\ell(N-K)} 2^{2K+Nm}} \quad .
 \end{aligned}$$

Theorem 4.4 There exists at least one non-systematic periodic code such that

$$\lim_{m \rightarrow \infty} \frac{\frac{d}{\text{FREE}}}{n_A} \geq \frac{R(1-2^{R-1})}{H(1-2^{R-1})+R-1} \quad .$$

Proof Note that no information sequence with a string of m or more all-zero blocks can produce the minimum free weight codeword. Hence if

$$\sum_{\ell=1}^{\ell_{\max}} F(\ell, d) < 1 ,$$

there exists at least one code with $d_{\text{FREE}} > d$. Let $F_{\max} = \max_{1 \leq \ell \leq \ell_{\max}} F(\ell, d)$. Then if $\ell_{\max} F_{\max} < 1$, there exists at least one code with $d_{\text{FREE}} > d$. Alternatively, if d_{FREE} is the smallest integer such that $\ell_{\max} F_{\max} \geq 1$, then there exists at least one code with free distance greater than or equal to d_{FREE} .

First an upper bound on F_{\max} will be obtained. Since

$$F(\ell, d) \leq \frac{T(2^K - 1)^2}{2^{2K+Nm}} \cdot \frac{2^{N(m+\ell)} H\left(\frac{d}{N(m+\ell)}\right)}{2^{\ell(N-K)}} . \quad \text{Therefore an upper bound on } F_{\max} \text{ can be obtained by maximizing } \left[N(m+\ell) H\left(\frac{d}{N(m+\ell)}\right) - \ell(N-K) \right] . \quad \text{Let } L \text{ be the value of } \ell \text{ which maximizes this expression. By setting the derivative of } \left[N(m+\ell) H\left(\frac{d}{N(m+\ell)}\right) - \ell(N-K) \right] \text{ equal to zero and solving for } \ell, \text{ it can be shown that } L = \frac{d}{N(1-2^{R-1})} - m . \quad \text{Therefore}$$

$$F_{\max} \leq \frac{T(2^K - 1)^2}{2^{2K+Nm}} \cdot \frac{2^{\frac{d}{1-2^{R-1}} H(1-2^{R-1})}}{\frac{d(N-K)}{N(1-2^{R-1})} - m(N-K)} .$$

Therefore if

$$\frac{\ell_{\max} T (2^K - 1)^2}{2^{2K + Nm}} \cdot \frac{\frac{d}{1-2^{R-1}} H(1-2^{R-1})}{\frac{d(N-K)}{N(1-2^{R-1})} - m(N-K)} < 1 ,$$

then $\ell_{\max} F_{\max} < 1$. Hence if d_{FREE} is the least integer such that

$$\frac{\ell_{\max} T (2^K - 1)^2}{2^{2K + Nm}} \cdot \frac{\frac{d_{\text{FREE}}}{1-2^{R-1}} H(1-2^{R-1})}{\frac{d_{\text{FREE}}(N-K)}{N(1-2^{R-1})} - m(N-K)} \geq 1 ,$$

then there exists at least one code with free distance greater than or equal to d_{FREE} .

Therefore the least integer d_{FREE} must be found such that

$$\begin{aligned} & \frac{(M+\ell_{\max})\ell_{\max}(2^K - 1)^2}{2} \cdot \frac{\frac{d_{\text{FREE}}}{1-2^{R-1}} H(1-2^{R-1}) - \frac{d_{\text{FREE}}(N-K)}{N(1-2^{R-1})}}{2^{m(N-K)}} \\ & \cdot 2^{m(N-K)} \geq 2^{2K+Nm} \quad \xrightarrow{\text{d}_{\text{FREE}}} \\ & 2 \log_2 \left[(m + \ell_{\max}) \ell_{\max} \right] + 2 \log_2 (2^K - 1) + \frac{d_{\text{FREE}}}{1-2^{R-1}} H(1-2^{R-1}) \\ & \cdot 2^{-\frac{d_{\text{FREE}}(N-K)}{N(1-2^{R-1})}} + m(N-K) \geq 2^{2K+Nm} \quad \xrightarrow{\text{d}_{\text{FREE}}} \\ & \log_2 \left[(m + \ell_{\max}) \ell_{\max} \right] + 2 \log_2 (2^K - 1) + m(N-K) - \frac{(1-R)d_{\text{FREE}}}{(1-2^{R-1})} \\ & + \frac{d_{\text{FREE}}}{(1-2^{R-1})} H(1-2^{R-1}) \geq 2K + Nm \quad \xrightarrow{\text{d}_{\text{FREE}}} \end{aligned}$$

$$\frac{\log_2 \left[(m+\ell_{\max}) \ell_{\max} \right]}{m} + \frac{2 \log_2 (2^K - 1)}{m} + (N-K) + \frac{d_{\text{FREE}}}{m(1-2^{R-1})}$$

$$\left[H(1-2^{R-1}) - (1-R) \right] \geq \frac{2K}{m} + N \longrightarrow$$

$$(N-K) + \frac{d_{\text{FREE}}}{m} \left[\frac{H(1-2^{R-1}) - (1-R)}{(1-2^{R-1})} \right] \geq N \quad \text{as } m \rightarrow \infty \longrightarrow$$

$$\frac{d_{\text{FREE}}}{m} \left[\frac{H(1-2^{R-1}) + R-1}{(1-2^{R-1})} \right] \geq K \quad \text{as } m \rightarrow \infty \longrightarrow$$

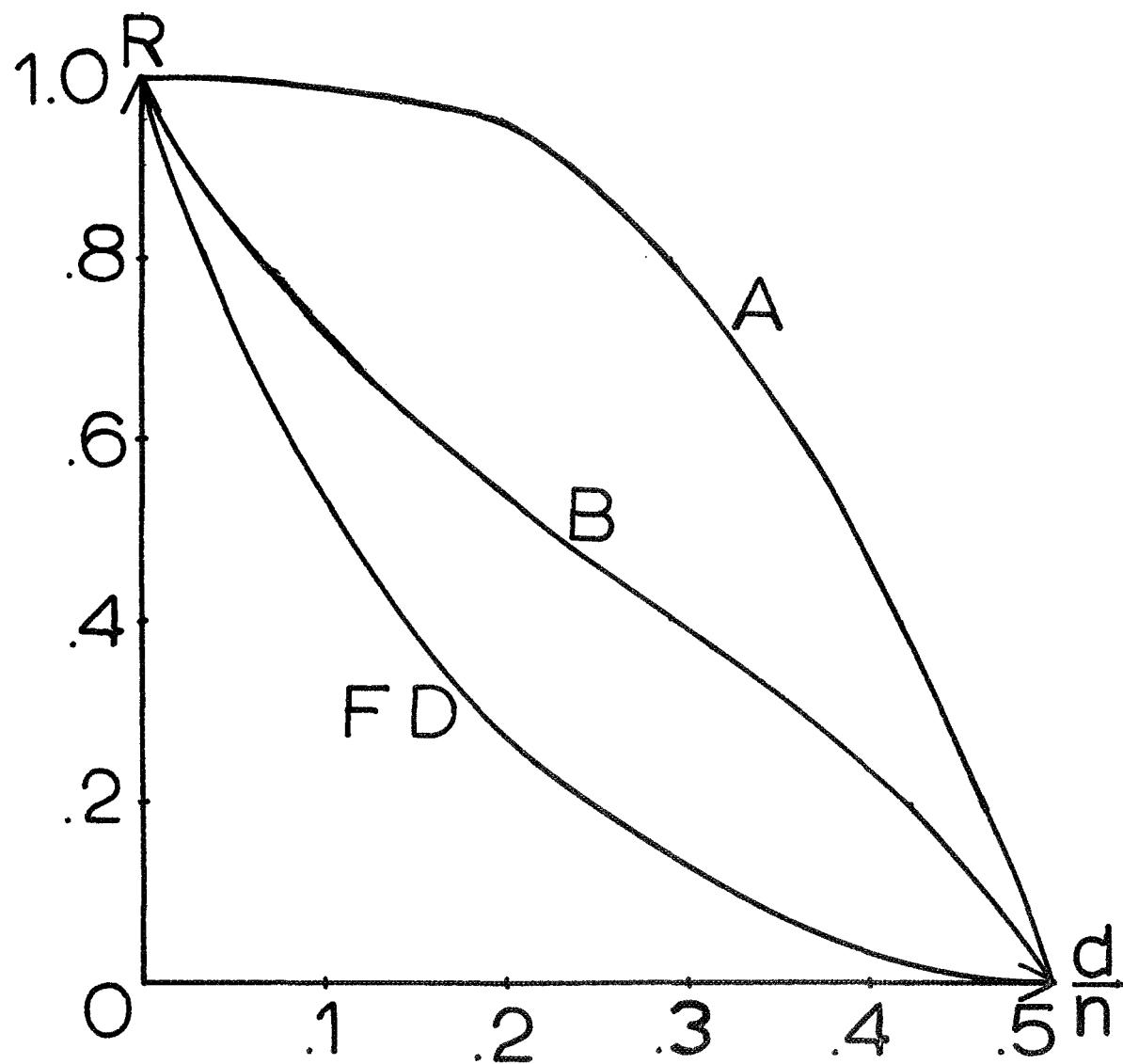
$$\frac{d_{\text{FREE}}}{m} \geq \frac{K(1-2^{R-1})}{H(1-2^{R-1}) + R-1} \quad \text{as } m \rightarrow \infty \longrightarrow$$

$$\lim_{m \rightarrow \infty} \frac{d_{\text{FREE}}}{n_A} \geq \frac{R(1-2^{R-1})}{H(1-2^{R-1}) + R-1} \quad . |$$

This bound is plotted in Figure 4.5 along with Neumann's bound and the usual Gilbert lower bound on d_{FD} . It is interesting to note that the bound given in theorem 4.4 is exactly the same as Neumann's bound for $R \leq 0.37$ with $2R$ replaced by R .

2. An Upper Bound on Error Probability for Maximum Likelihood Decoding over a BSC

Viterbi [2] has given upper and lower bounds on the reliability function, $E(R)$, for the best periodic code used with maximum likelihood decoding over a BSC. Theorem 4.4 can be used to obtain a lower bound on $E(R)$, i.e., an upper bound on error probability, which is better than Viterbi's bound for low rates and which meets Viterbi's upper bound on $E(R)$ at $R = 0$.



A = bound of theorem 4.4

B = bound of equation (110)

Fig. 4.5. A comparison of lower bounds on d_{FREE} .

Assume that the length of the information sequence is ℓ and let the period of the code be T . There are then $2^{K\ell}$ possible transmitted sequences beginning at time u , $0 \leq u < T$. Now label the non-zero codewords beginning at each time u from 1 to $2^{K\ell} - 1$ and let $w_i^u(u)$ be the Hamming weight of the i^{th} non-zero codeword beginning at time u , $0 \leq u < T$.

Gallager [25] has shown that the probability of error for an $R = k/n$ block code used with maximum likelihood decoding over a BSC with digit error probability p is bounded by

$$P_e \leq \sum_{i=1}^{2^k - 1} e^{-[w_i^u \log_e p']} , \quad (112)$$

where $p' = 2 \sqrt{p(1-p)}$ and w_i^u is the Hamming weight of the i^{th} non-zero codeword. Hence for periodic convolutional codes this bound becomes

$$P_e \leq T \sum_{i=1}^{2^K - 1} e^{-[w_i^u \log_e p']} ; \quad (113)$$

where $w_i^u = \min_{0 \leq u < T} \{ w_i^u(u) \}$. Equation (113) can be rewritten as

$$P_e \leq T \sum_{j=0}^{\ell-1} 2^{jk} e^{-w_j \log_e p'} , \quad (114)$$

where w_j is a lower bound on the Hamming weight of all the non-zero codewords over their last $m+j+1$ blocks. Note that the first $\ell+m-m-j-1 = \ell-j-1$ blocks of each non-zero codeword agree with the first $\ell-j-1$ blocks of the transmitted codeword, and hence the distance contribution of these blocks is

omitted in (114).

Since the best code satisfies any lower bound on minimum distance, the weight of the last $m+j+1$ blocks of each non-zero codeword in the best code can be underbounded by both the usual Gilbert lower bound on d_{FD} (which also applies to d_{FREE} since $d_{FD} \leq d_{FREE}$) and for large enough T by the lower bound on d_{FREE} given in theorem 4.4. Hence

$$w_j = \max \left[\lambda' N(m+1), \lambda N(j+m+1) \right], \quad (115)$$

$$\text{where } \lambda' = \frac{d_{FREE}}{N(m+1)} = \frac{R(1-2^{R-1})}{H(1-2^{R-1}) + R - 1} \text{ and } \lambda = \frac{d_{FREE}}{N(m+1)} = H^{-1}(1-R).$$

Let j_0 be the value of j at which $\lambda' N(m+1) = \lambda N(j+m+1)$. Then for $j < j_0$, $\lambda' N(m+1)$ is the dominant term in (115) and for $j \geq j_0$, $\lambda N(j+m+1)$ is the dominant term in (115). Let $Q = \frac{j_0}{m}$.

Theorem 4.5 For maximum likelihood decoding over a BSC,

$$\lim_{m \rightarrow \infty} P_e < e^{-n_A} \begin{bmatrix} -QR - \lambda' \log_e p' \\ \end{bmatrix}$$

for the best periodic code with large enough T , if $K \log_e 2 + \lambda' N \log_e p'$

$$\frac{\lambda' N \log_e p'}{Q+1} < 0.$$

Proof

$$P_e \leq T \sum_{j=0}^{\ell-1} 2^{jk} e^{w_j \log_e p'} \longrightarrow$$

$$P_e \leq e^{\log_e T} \left[\sum_{j=0}^{j_0-1} 2^{jk} e^{\lambda' n_A \log_e p'} + \sum_{j=j_0}^{\ell-1} 2^{jk} e^{\lambda N(j+m+1) \log_e p'} \right] \rightarrow$$

$$P_e < e^{\log_e T} \left[2^{j_0 K} e^{\lambda' n_A \log_e p'} + \sum_{i=0}^{\ell-j_0-1} 2^{(j_0+i)K} \right]$$

$$P_e < e^{\log_e T} \left[e^{j_0 NR'} e^{\lambda' n_A \log_e p'} + \sum_{i=0}^{\ell-j_0-1} e^{(j_0+i)NR'} \right. \\ \left. e^{\lambda N(j_0+i+m+1) \log_e p'} \right],$$

where $R' = R \log_e 2$ is in nats. Since $\lambda' N(m+1) = \lambda N(j_0+m+1)$,

$$P_e < e^{\log_e T} \left[e^{j_0 NR'} + \lambda' n_A \log_e p' + \sum_{i=0}^{\ell-j_0-1} e^{(j_0+i)NR'} \right. \\ \left. e^{\lambda' n_A (1 + \frac{i}{j_0+m+1}) \log_e p'} \right] \longrightarrow$$

$$P_e < e^{\log_e T} + j_0 NR' + \lambda' n_A \log_e p' \left[1 + \sum_{i=0}^{\ell-j_0-1} e^{iNR'} \right]$$

$$e^{\lambda' n_A (\frac{i}{j_0+m+1}) \log_e p'} \longrightarrow$$

$$P_e < e^{-n_A} \left(-\frac{\log_e T}{n_A} - \frac{j_0}{m+1} R' - \lambda' \log_e p' \right) \left[1 + \sum_{i=0}^{\ell+j_0-1} \right]$$

$$e^{i(NR' + \frac{\lambda' n_A}{j_0+m+1} \log_e p')} \longrightarrow$$

$$\lim_{m \rightarrow \infty} P_e < e^{-n_A} (-QR' - \lambda' \log_e p') \left[1 + \sum_{i=0}^{\ell-j_0-1} \right]$$

$$e^{i(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')}] .$$

If $(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p') < 0$,

$$\lim_{m \rightarrow \infty} P_e < e^{-n_A} (-QR' - \lambda' \log_e p') \left[1 + \sum_{i=0}^{\infty} \right]$$

$$e^{i(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')}] .$$

$$\text{But } \sum_{i=0}^{\infty} e^{i(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')} = 1 + \sum_{i=1}^{\infty}$$

$$\left[e^{(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')} \right]^i$$

when $(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p') < 0$. Hence

$$\sum_{i=0}^{\infty} e^{i(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')} = 1 + \frac{1}{1 - e^{(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')}}.$$

by the geometric series argument. Therefore

$$\lim_{m \rightarrow \infty} P_e < e^{-n_A (-Q R' - \lambda' \log_e p')} \left[2 + \frac{1}{1 - e^{(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')}} \right].$$

Let $\left[2 + \frac{1}{1 - e^{(K \log_e 2 + \frac{\lambda' N}{Q+1} \log_e p')}} \right] = B$, a positive constant

depending only on R and p ($Q \approx \frac{\lambda'}{\lambda} - 1$ for large m), where

$3 \leq B < \infty$. Then

$$\lim_{m \rightarrow \infty} P_e < e^{-n_A \left(-\frac{\log B}{n_A} - Q R' - \lambda' \log_e p' \right)} \quad \longrightarrow$$

$$\lim_{m \rightarrow \infty} P_e < e^{-n_A (-Q R' - \lambda' \log_e p')} . |$$

Corollary 4.3 For maximum likelihood decoding over a BSC,

$$\lim_{m \rightarrow \infty} E(R) > -Q R' - \lambda' \log_e p'$$

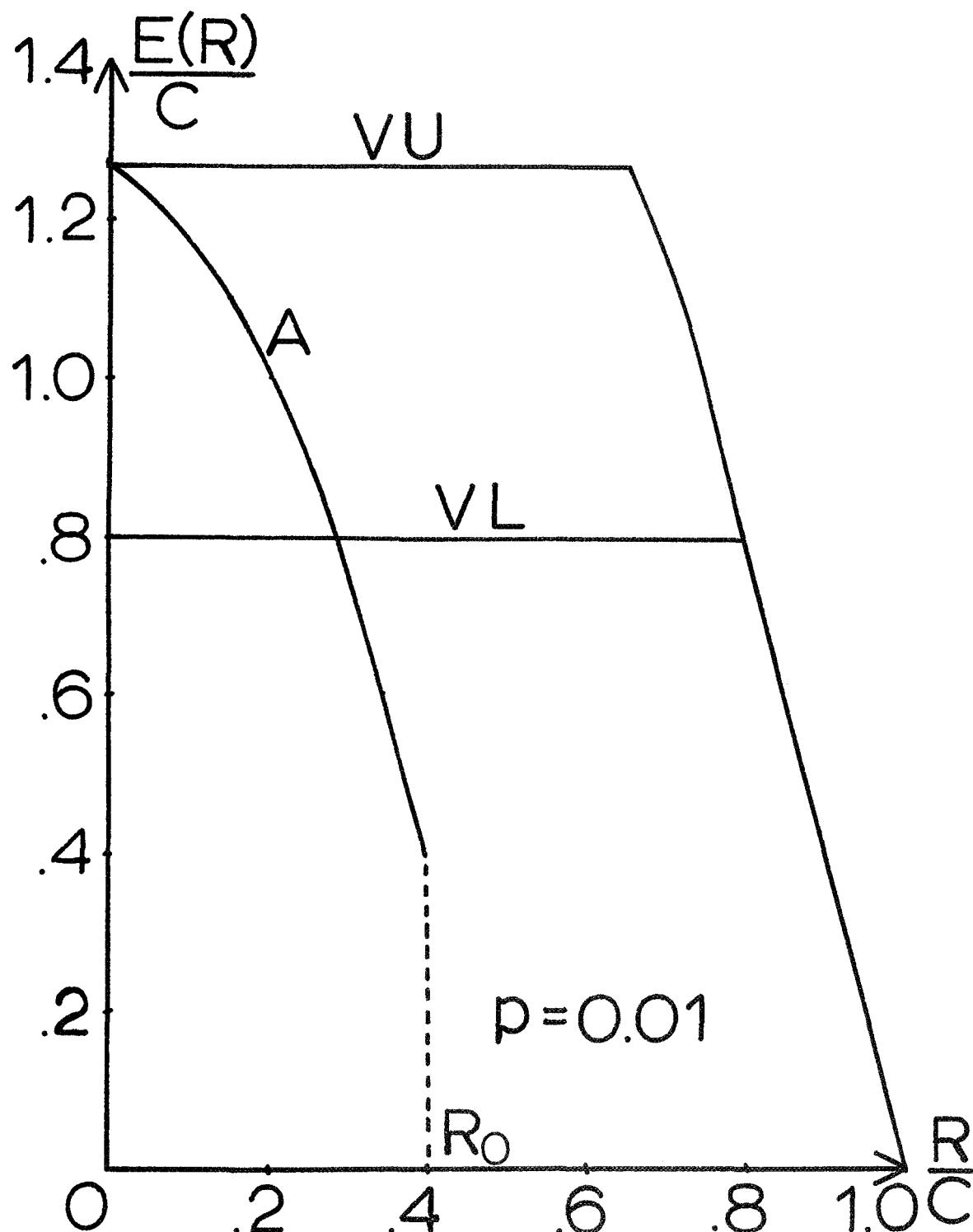
if $\lambda' N \log_e p'$

$$K \log_e 2 + \frac{\lambda' N \log_e p'}{Q+1} < 0, \text{ where } E(R) = -\frac{1}{n_A} \log_e P_e \text{ is}$$

the reliability function for the best periodic code with
large enough T . |

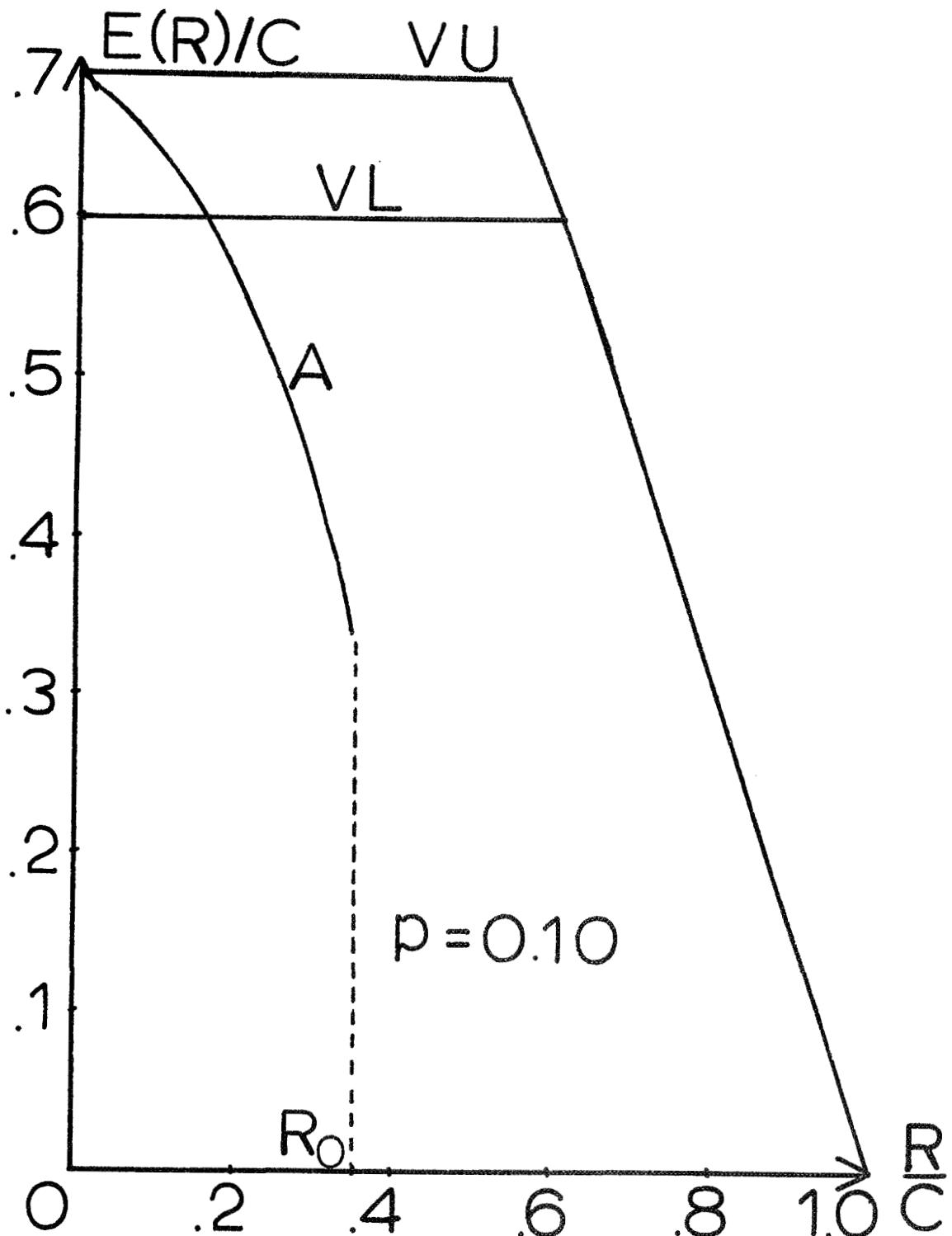
$$\text{Let } R_0 \text{ be the value of } R \text{ such that } K \log_e 2 + \frac{\lambda' N \log_e p'}{Q+1} = 0.$$

Figures 4.6, 4.7, and 4.8 compare the bound of corollary 4.3
with Viterbi's upper and lower bound on $E(R)$ for all $R < R_0$,



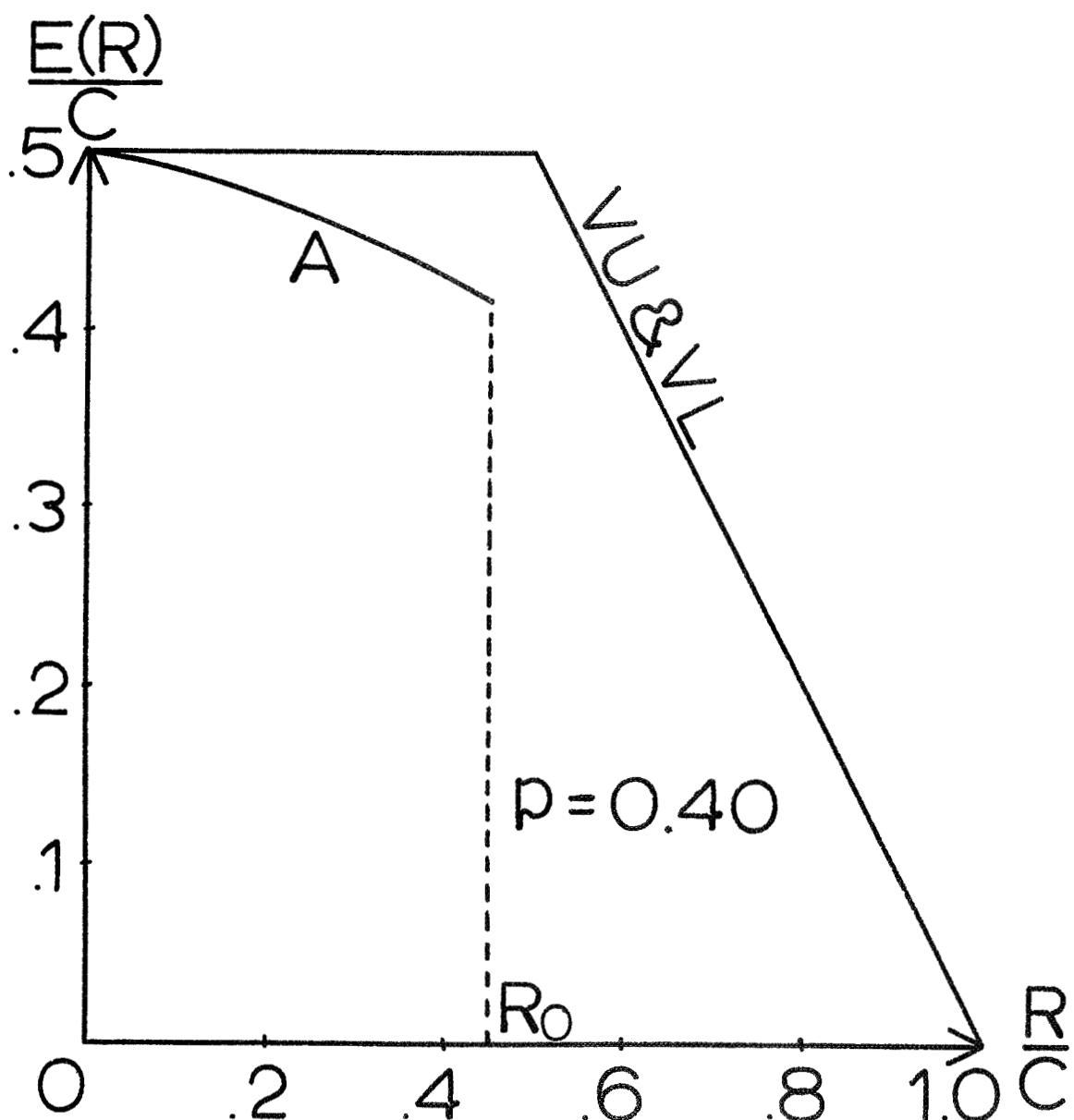
A = bound of corollary 4.3.

Fig. 4.6 A comparison of the bound of corollary 4.3 with Viterbi's upper bound (VU) and Viterbi's lower bound (VL) for $p = 0.01$.



Δ = bound of corollary 4.3.

Fig. 4.7. A comparison of the bound of corollary 4.3 with Viterbi's upper bound (VU) and Viterbi's lower bound (VL) for $p = 0.10$.



A = bound of corollary 4.3.

Fig. 4.8. A comparison of the bound of corollary 4.3 with Viterbi's upper bound (VU) and Viterbi's lower bound (VL) for $p = 0.40$.

with $p = 0.01, 0.10$, and 0.40 , respectively. In each of

these figures $\frac{R}{C}$ is plotted against $\frac{E(R)}{C}$, where $C = 1+p \log_2$

$p + (1-p) \log_2 (1-p) = 1 + H(p)$ is the capacity of a BSC.

Table 4.1 gives values of R_0 , C , and $\frac{R_0}{C}$ for $p = 0.01, 0.10$, and 0.40 .

It is easy to see from Figures 4.6 and 4.7 that the bound of corollary 4.3 is superior to Viterbi's lower bound on $E(R)$ for low rates and for low values of p . For higher values of p , near $p = 0.50$, Viterbi's lower bound on $E(R)$ coincides with his upper bound on $E(R)$, and hence no improvement is possible, as can be seen from Figure 4.8. Note also that the bound of corollary 4.3 meets Viterbi's upper bound on $E(R)$ at $R = 0$. This can be shown analytically as follows.

Let $E_u(R)$ be Viterbi's upper bound on $E(R)$ and let $E_L(R)$ be the lower bound on $E(R)$ of corollary 4.3. Then

$$E_L(R) \Big|_{R=0} = -\frac{1}{2} \log_e 2 \sqrt{p(1-p)} . \quad (116)$$

$$E_u(R) \Big|_{R=0} = -\lim_{s \rightarrow \infty} \left\{ s \log_e \left[\frac{1}{2} \left(\sqrt{p^2} + \sqrt{(1-p)^2} \right)^{1/s} \right] \right\} \quad (117)$$

$$= -\lim_{s \rightarrow \infty} \left\{ s \log_e \left[\frac{1}{2} + \frac{1}{2} \left(2 \sqrt{p(1-p)} \right)^{1/s} \right] \right\} \quad (118)$$

$$= -\lim_{s \rightarrow \infty} s \left\{ \log_e \frac{1}{2} + \log_e \left[1 + \left(2 \sqrt{p(1-p)} \right)^{1/s} \right] \right\} \quad (119)$$

$$= -\lim_{s \rightarrow \infty} \frac{s \log_e \left[1 + \left(2 \sqrt{p(1-p)} \right)^{1/s} \right]}{2} \quad (120)$$

$$= -\lim_{s \rightarrow \infty} \frac{\log_e \frac{1}{2} + \log_e \left[1 + \left(2 \sqrt{p(1-p)} \right)^{1/s} \right]}{1/s} . \quad (121)$$

TABLE 4.1

	$p = 0.01$	$p = 0.10$	$p = 0.40$
R_0	.37 bits	.19 bits	.013 bits
C	.92 bits	.53 bits	.03 bits
$\frac{R_0}{C}$.40	.35	.45

Applying L'Hospital's rule,

$$E_u(R) \Big|_{R=0} = - \frac{\frac{1}{\left[1 + \left(2\sqrt{p(1-p)}\right)^{1/s}\right]} \cdot \left(2\sqrt{p(1-p)}\right)^{1/s}}{\frac{1}{s^2}} \cdot \frac{\log_e\left(2\sqrt{p(1-p)}\right) \cdot 1/s^2}{\frac{1}{s^2}} \quad |_{s=\infty} \quad (122)$$

$$= - \frac{\left(2\sqrt{p(1-p)}\right)^{1/s}}{\left[1 + \left(2\sqrt{p(1-p)}\right)^{1/s}\right]} \cdot \log_e\left(2\sqrt{p(1-p)}\right) \Big|_{s=\infty} \quad (123)$$

$$= -1/2 \log_e\left(2\sqrt{p(1-p)}\right). \quad (124)$$

Therefore $E_u(0) = E_L(0)$.

3. Upper Bounds

McEliece and Rumsey [18] have shown that for $R = \frac{1}{N}$ systematic binary fixed codes

$$\frac{d_{\text{FREE}}}{n_A} < \frac{1-R}{2} + \frac{\log_2 n_E}{2(m+1)} + \frac{1}{n_A} \quad (125)$$

if $\frac{n_E}{1+\log_2 n_E} > N$, where $n_E = (N-1)(m+1)$ is the number of parity digits in one constraint length of transmitted digits.

It can easily be shown that $\frac{n_E}{1+\log_2 n_E} > N$ for some finite value of m . For instance, if $R = 1/2$, (125) holds for all $m \geq 8$.

Hence

$$\lim_{m \rightarrow \infty} \frac{d_{\text{FREE}}}{n_A} < \frac{1-R}{2} \quad (126)$$

for all $R = \frac{1}{N}$ systematic binary fixed codes. It can be shown that this result extends to systematic binary fixed codes of all rates. Equation (125) then becomes

$$\frac{d_{\text{FREE}}}{n_A} < \frac{1-R}{2} + \frac{\log_2 n_E}{2K(m+1)} + \frac{(K+1)}{2n_A} \quad \text{for } (N-K) \geq 2 \quad (127)$$

$$\frac{d_{\text{FREE}}}{n_A} < \frac{1-R}{2} + \frac{\log_2 n_E}{2K(m+1)} + \frac{(K+2)}{2n_A} \quad \text{for } (N-K) = 1 \quad (128)$$

if $\frac{n_E}{K + \log_2 n_E} > \frac{1}{R}$, where $n_E = (N-K)(m+1) = n_A(1-R)$.

Again equation (126) results in the limit as $m \rightarrow \infty$.

In order to extend this result to non-systematic binary fixed codes, it is convenient to make the following definitions.

Definition 4.1 $M_j = \max_{1 \leq i \leq K} \left\{ \text{degree} \left[G_i^{(j)}(D) \right] \right\}, \quad 1 \leq j \leq N,$

for those fixed codes whose matrix of generator functions $G(D)$ contains only polynomial elements. |

M_j is called the constraint span of the j^{th} encoded sequence.

Note that $m = \max_{1 \leq j \leq N} \left[\sum_{j=1}^N M_j \right]$.

Definition 4.2 $n_E = \sum_{j=1}^N M_j + N$ is the number of transmitted digits that can be affected by a non-zero information block x_0 for fixed codes with constraint spans $M_j, j = 1, \dots, N$. |

This definition of n_E is slightly different from the definition given by McEliece and Rumsey.

Theorem 4.6 For any fixed convolutional code,

$$d_{\text{FREE}} < \frac{n_E}{2} + \frac{1}{2R} \log_2 n_E + \frac{1}{2} \quad \text{if } \frac{n_E}{K + \log_2 n_E} > \frac{1}{R} .$$

Proof Consider all information sequences of length h . The average weight of a codeword is

$$\frac{Nh + \sum_{j=1}^N M_j}{2}.$$

Since d_{FREE} is less than or equal to the average weight of all non-zero codewords,

$$d_{\text{FREE}} \leq \frac{2^{Kh}}{2^{Kh-1}} \left(\frac{Nh + \sum_{j=1}^N M_j}{2} \right)$$

or

$$d_{\text{FREE}} \leq \frac{Nh + \sum_{j=1}^N M_j}{2} + \frac{1}{2^{Kh-1}} \left(\frac{Nh + \sum_{j=1}^N M_j}{2} \right).$$

Choose h such that $\frac{Nh}{2} + \frac{n_E - N}{2} < 2^{Kh-1}$. Then $d_{\text{FREE}} \leq \frac{Nh}{2} + \frac{n_E - N}{2} + \frac{1}{2}$.

Now a more explicit way of choosing h will be derived. Suppose h is chosen such that $2^{K(h-1)} < n_E \leq 2^{Kh}$. Then

$$\frac{Nh}{2} + \frac{n_E - N}{2} \leq \frac{Nh}{2} + \frac{2^{Kh} - N}{2}.$$

If $\frac{n_E}{K + \log_2 n_E} > \frac{1}{R}$, then $\frac{2^{Kh}}{Kh} > \frac{n_E}{K + \log_2 n_E} > \frac{1}{R}$ since $Kh - K < \log_2 n_E$. This implies that $h < \frac{2^{Kh}}{N}$. Therefore

$$\frac{Nh}{2} + \frac{n_E - N}{2} < 2^{Kh-1} + 2^{Kh-1} - \frac{N}{2} = 2^{Kh} - \frac{N}{2} \leq 2^{Kh-1},$$

since $N \geq 2$ except for the trivial $R = \frac{1}{1} = 1$ codes.

Consequently, for $\frac{n_E}{K + \log_2 n_E} > \frac{1}{R}$, if h is chosen such

that $2^{K(h-1)} < n_E \leq 2^{Kh}$, it is also chosen such that

$$\frac{Nh}{2} + \frac{n_E - N}{2} < 2^{Kh-1}.$$

Finally, since $h < \frac{K + \log_2 n_E}{K}$,

$$d_{\text{FREE}} < \frac{N}{2K} (K + \log_2 n_E) + \frac{n_E - N}{2} + \frac{1}{2} = \frac{n_E}{2} + \frac{\log_2 n_E}{2R} + \frac{1}{2} . |$$

Corollary 4.4 $\lim_{m \rightarrow \infty} d_{\text{FREE}} < \frac{n_E}{2} .$

Proof The last two factors in theorem 4.6 become negligible compared with $\frac{n_E}{2}$ as $m \rightarrow \infty$. Also, it can be shown

that

$$\frac{n_E}{K + \log_2 n_E} > \frac{1}{R} \quad \text{for all } R \text{ as } m \rightarrow \infty . |$$

Note that theorem 4.6 and corollary 4.4 are completely general, i.e., they apply to both systematic and non-systematic binary fixed codes of all rates. In the systematic case, these bounds are exactly the same as the bounds of (125) and (126) except for the slightly different definition of n_E .

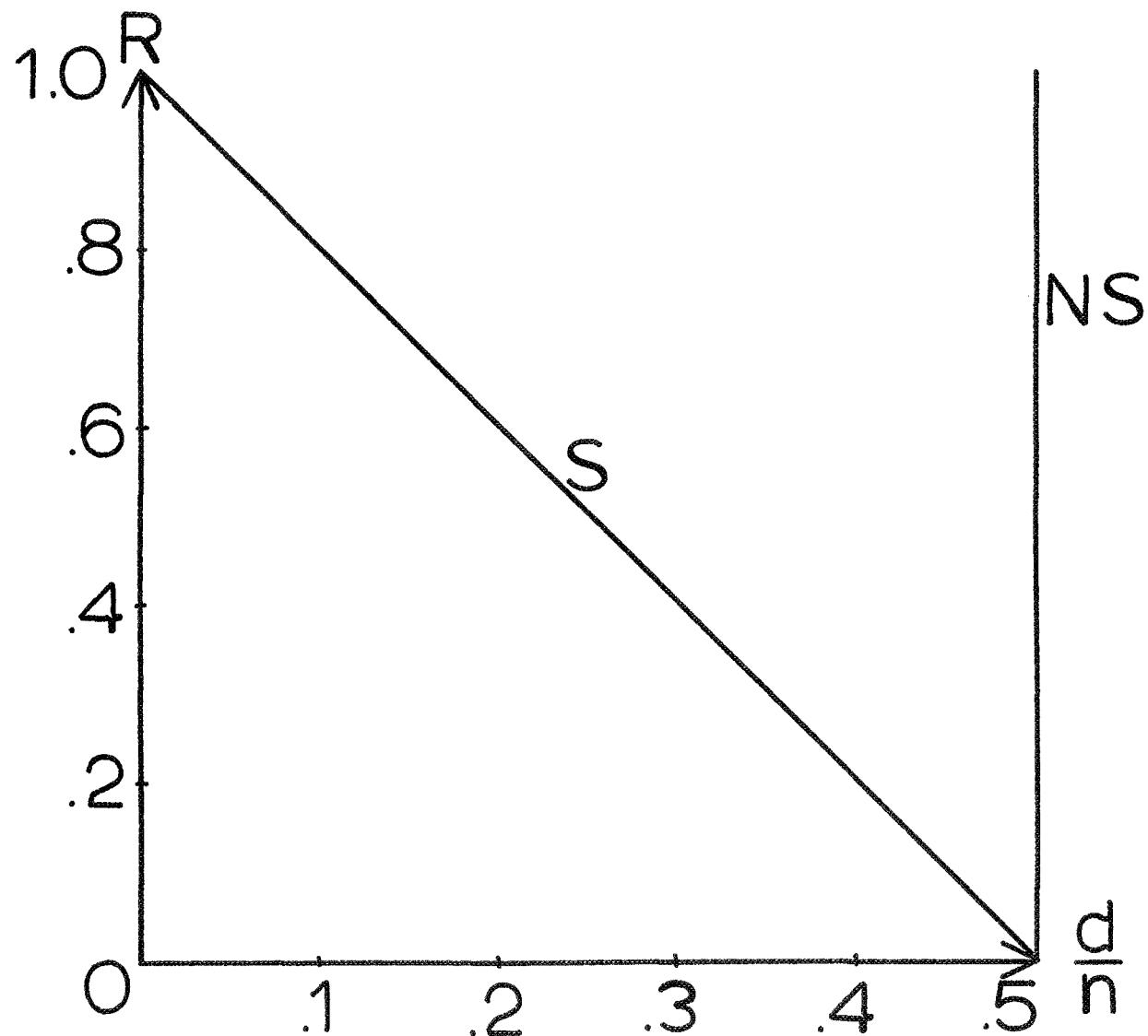
Note that for systematic codes, $M_1 = M_2 = \dots = M_K = 0$.

Hence corollary 4.4 indicates that more free distance may be available for non-systematic codes than for systematic codes of the same memory. The bound of corollary 4.4 for non-systematic codes is shown together with the bound of equation (126) for systematic codes in Figure 4.9.

For periodic codes, the constraint spans M_j must be re-defined as

$$M_j = \max_{0 \leq u < T} \max_{0 \leq i \leq m} \left\{ i \mid \text{the } j^{\text{th}} \text{ column of } G_i(u) \neq 0 \right\} . \quad (129)$$

Again $m = \max_{1 \leq j \leq N} [M_j]$. Then by a slight modification of the proofs, theorem 4.6 and corollary 4.4 can be shown to hold for periodic codes.



S = systematic bound

NS = non-systematic bound

Fig. 4.9. A comparison of upper bounds on d_{FREE} for systematic and non-systematic codes.

Figure 4.10 sums up the results of Section IV.C. by plotting together the usual Gilbert lower bound on d_{FD} , and the bounds of equation (110), theorem 4.4, equation (126), and corollary 4.4. Note that the lower bounds on d_{FREE} of equation (110) and theorem 4.4 for non-systematic codes cross the upper bound on d_{FREE} of equation (126) for systematic codes. This fact guarantees that more free distance is available with non-systematic codes than with systematic codes.

Example 4.1 For $R = 1/2$, $m = 3$, the best fixed systematic code, viz. the code with $G^{(1)}(D) = 1$, $G^{(2)}(D) = 1 + D + D^3$, has $d_{FREE} = 4$. For this code $M_1 = 0$, $M_2 = 3$, and $n_E = 5$. Now consider the $R = 1/2$ non-systematic fixed code with $m = 3$ and $G^{(1)}(D) = 1 + D + D^3$, $G^{(2)}(D) = 1 + D^2 + D^3$. This code has $d_{FREE} = 6$, $M_1 = 3$, $M_2 = 3$, and $n_E = 8$. Hence for $m = 3$, non-systematic codes are clearly superior to systematic codes. |

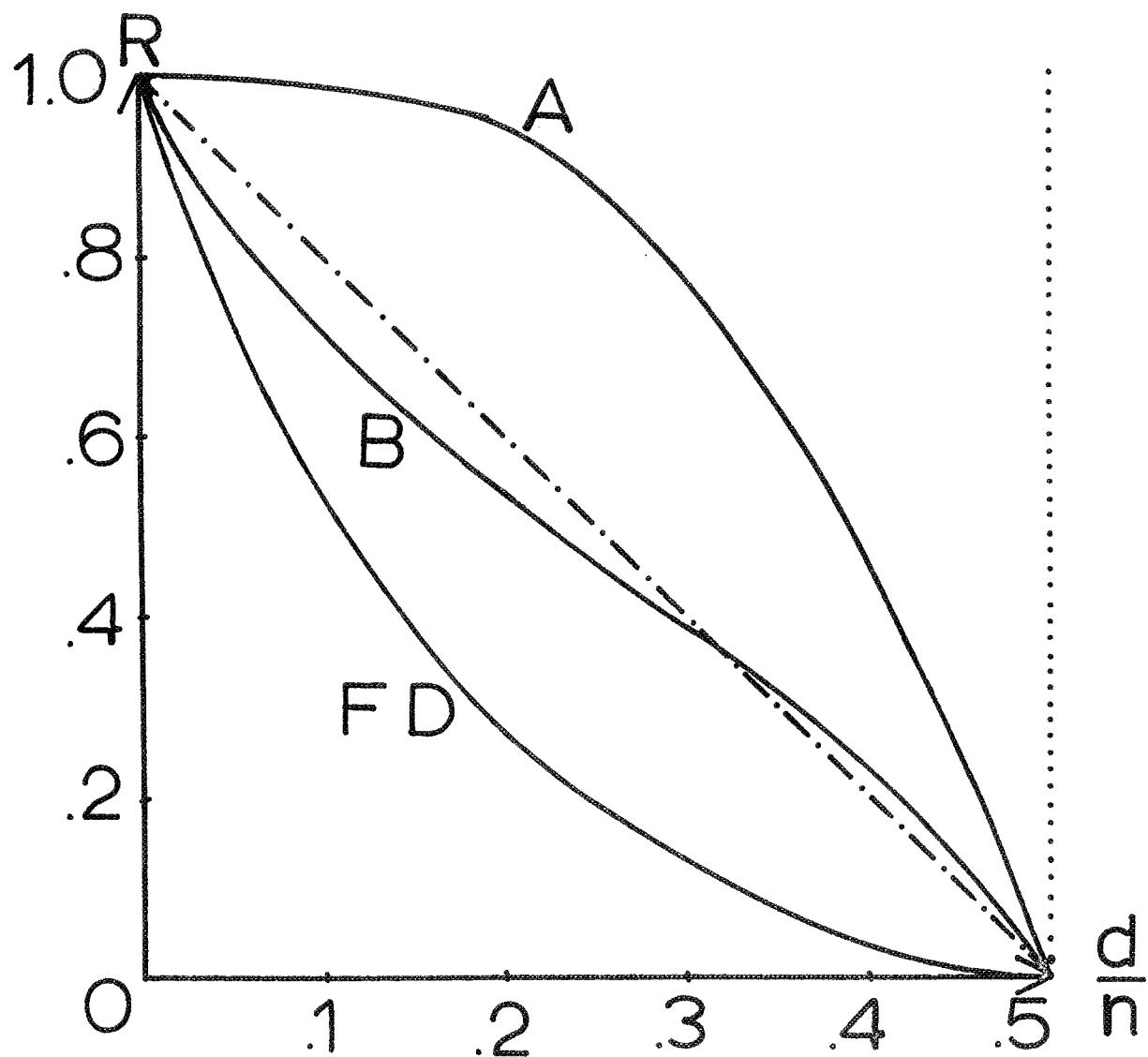
Example 4.2 In this example a fixed code which meets the bound of theorem 4.6 will be presented. For $R = 1/2$, $m = 4$, let $G^{(1)}(D) = 1 + D + D^2 + D^4$, $G^{(2)}(D) = 1 + D^2 + D^3 + D^4$.

Then

$$n_E = \sum_{j=1}^2 M_j + 2 = 10 \text{ and } \frac{n_E}{K + \log_2 n_E} = \frac{10}{1 + \log_2 10} > \frac{1}{R} = 2,$$

so the bound is valid and yields $d_{FREE} < \frac{n_E}{2} + \frac{1}{2R} \log_2 n_E + 1/2 = 5 + \log_2 10 + 1/2 = 8.8$, which implies that $d_{FREE} \leq 8$.

It can easily be shown that $d_{FREE} = 8$ for the above code. |



A = bound of theorem 4.4.

B = bound of equation (110)

_____ = bound of equation (126) = bound of corollary 4.4

Fig. 4.10. A comparison of bounds on d_{FREE} .

The fact that non-systematic codes can produce more free distance than systematic codes is important when sequential decoding is being used. Recall that d_{FREE} is a more appropriate distance measure for sequential decoding than d_{FD} or d_{DD} . Hence a sequential decoder should exhibit a lower probability of decoding error for the best non-systematic code of a given memory than for the best systematic code of the same memory. Happily, Bucher [26] has obtained theoretical results which indicate the possibility of achieving lower error probabilities with non-systematic codes than with systematic codes. Experimental verification of this fact is given in Chapter VI.

E. A Gilbert Lower Bound for an Easily Instrumented Sub-class of $R = 1/2$ Non-Systematic Codes

The usual Gilbert lower bound argument states that there must exist at least one code with $d_{\text{FD}} \geq d$ if d is the least integer such that

$$\sum_{j=0}^d \left[\begin{array}{l} \text{no. of codewords with} \\ \underline{x}_0 \neq 0 \text{ of weight } j \end{array} \right] \times \left[\begin{array}{l} \text{no. of codes which can} \\ \text{generate a particular} \\ \text{codeword of weight } j \end{array} \right] \geq \left[\begin{array}{l} \text{total no.} \\ \text{of codes} \end{array} \right]. \quad (130)$$

Consider the class of $R = 1/2$ fixed non-systematic codes with $G^{(2)}(D) = D + G^{(1)}(D)$ and $g_0^{(1)} = g_0^{(2)} = 1$. For any information sequence and any $G^{(1)}(D)$ with $g_0^{(1)} = 1$, each

codeword \underline{y} always has exactly one 1 in y_1 . Hence the number of first constraint length code digits which can be chosen independently is $n_A - 1 = 2m + 1$.

For any information sequence \underline{x} whose transform is $x(D)$, the sum of the transforms of the two transmitted sequences is given by

$$x(D) G^{(1)}(D) + x(D) [D + G^{(1)}(D)] = y^{(1)}(D) + y^{(2)}(D). \quad (131)$$

Hence

$$Dx(D) = y^{(1)}(D) + y^{(2)}(D) \quad (132)$$

and

$$x_j = y_{j+1}^{(1)} + y_{j+1}^{(2)}, \quad j = 0, 1, \dots. \quad (133)$$

Therefore a particular choice of $[y]_m$ fixes $[x]_{m-1}$ and x_m can be chosen arbitrarily. Also, for any particular choice of $[x]$ and $[y]$, the matrix of generator functions $G(D) = [G^{(1)}(D) \ D + G^{(1)}(D)]$ is fixed. Hence only two codes can produce any specified choice of $[y]$. Note also that there are 2^m possible choices for $G^{(1)}(D)$ and therefore 2^m possible codes. Equation (130) then reduces to

$$\sum_{j=0}^{d-1} \binom{2m}{j} \times 2 \geq 2^m. \quad (134)$$

But

$$\sum_{j=0}^{d-1} \binom{2m}{j} \leq 2^{(2m)} H\left(\frac{d-1}{2m}\right) \text{ and (134) becomes } \\ (2m) H\left(\frac{d-1}{2m}\right) \geq m-1, \quad (135)$$

or

$$\lim_{m \rightarrow \infty} H\left(\frac{d}{n_A}\right) \geq \frac{1}{2}. \quad (136)$$

This proves the following result.

Theorem 4.7 For $R = 1/2$ non-systematic fixed codes with $G^{(2)}(D) = D + G^{(1)}(D)$, there exists at least one code such that

$$\lim_{m \rightarrow \infty} \frac{d_{FD}}{\frac{n}{A}} \geq .110.$$

Note that theorem 4.7 is exactly the same as the Gilbert lower bound on d_{FD} for $R = 1/2$. Therefore it is guaranteed that this class contains good codes. Since non-systematic codes are better for sequential decoding than systematic codes, codes of this type should perform very well. Another property of these codes is that they can be simply instrumented and that they possess the principal advantage of systematic codes, the so-called "quick look" capability. The instrumentation and "quick look" capability of these codes will be explained in detail along with their construction, simulation, and performance in Chapter VI.

V. Some Results on Free Distance

A. Bounding the Length of the Information Sequence Which Produces the Shortest Minimum Free Weight Codeword

Property F4 of free distance implies that a finite number of blocks of information digits are needed to produce the minimum free weight codeword for non-systematic fixed codes. And Lemma 4.2 shows that a finite number of blocks are needed for non-systematic periodic codes.

For the special case of $R = 1/2$ non-systematic fixed codes with $G^{(1)}(D)$ and $G^{(2)}(D)$ relatively prime polynomials an improved bound on the length of information sequence needed to produce the minimum free weight codeword can be obtained. Since $G^{(1)}(D)$ and $G^{(2)}(D)$ are relatively prime, there exist polynomials $A(D)$ and $B(D)$ of degree less than m such that $A(D) G^{(1)}(D) + B(D) G^{(2)}(D) = 1$ [27]. Hence for any information sequence \underline{x} whose transform is $x(D)$,

$$x(D) A(D) G^{(1)}(D) + x(D) B(D) G^{(2)}(D) = x(D) \left[A(D) G^{(1)}(D) + B(D) G^{(2)}(D) \right] \quad (137)$$

$$= x(D) . \quad (138)$$

Since an information sequence capable of producing the minimum free weight codeword cannot have any span of m consecutive zeros, the minimum free weight codeword cannot have any span of $2m-1$ consecutive all-zero blocks. Therefore at least one 1 must be produced in every $2m-1$ encoded blocks.

But every $R = 1/2$ non-systematic fixed code has d_{FREE} at most $2(m+1)$, and $2(m+1)$ 1's must be produced in the minimum free weight codeword within $(2m+1)(2m-1)+1$ blocks. Hence

$$d_{\text{FREE}} = r_{4m^2-m-1} . \quad (139)$$

Note that the bound of equation (139) is derived in a similar fashion to the bound of property F4 since the matrix

$$\underline{G}^{-1}(D) = \begin{bmatrix} A(D) \\ B(D) \end{bmatrix} \text{ is a zero-delay feedforward inverse for}$$

$\underline{G}(D)$ whose polynomial elements have maximum degree $M = m-1$.

A similar argument can be used to derive the following bound for fixed systematic codes of all rates:

$$d_{\text{FREE}} = r_{(N-K)(m+1)m} . \quad (140)$$

These bounds appear to be very weak since in practice the minimum free weight codeword is almost always produced by the first $m+1$ blocks of information digits. This remains an important problem since a tight bound would greatly simplify the calculation of d_{FREE} . The difficulty in proving a tight bound stems from our lack of knowledge about the weights of products of polynomials. However, for some special cases, tighter bounds can be obtained. For instance, if d'_{FD} is the reverse feedback decoding minimum distance and if

$$d_{FD} + d'_{FD} > r_i$$

for any i , then

$$d_{\text{FREE}} = r_{2m} . \quad (141)$$

This follows from the fact that any information sequence with some $x_i \neq 0$, $i > 2m$, produces a codeword with weight at least d_{FD} over the first $m+1$ transmitted blocks and weight at least d'_{FD} over the last $m+1$ transmitted blocks. Theorem 2.4 then yields (141). If the code is reversible, then (141) holds if $2d_{FD} > r_i$ for any i .

Also, for almost all fixed and periodic systematic codes, lemma 4.2 can be modified as follows:

$$d_{FREE} = r_{(3I)} m \quad , \quad (142)$$

where I is the least integer such that $r_i < d_{FD} + Id_{DD}$ for any i , d_{DD} satisfies equation (105) with equality in the fixed case and satisfies theorem 4.1 with equality in the periodic case, and d_{FD} satisfies equation (103) with equality.

As noted earlier, Neumann [11] has suggested that the correct bound on the length of information sequence needed to produce the minimum free weight codeword is $m+1$ blocks, i.e., $d_{FREE} = r_m$. Unfortunately, the following counterexample disproves this conjecture.

Example 5.1 Let $G^{(1)}(D) = 1 + D^2 + D^4 + D^5 + D^6 + D^{10} + D^{11}$ and $G^{(2)}(D) = 1 + D^2 + D^3 + D^4 + D^8 + D^9$ for an $R = 1/2$ non-systematic fixed code with $m = 11$. For $x(D) = 1 + D^2 + D^5 + D^8 + D^9 + D^{10} + D^{11} + D^{12}$, $y^{(1)}(D) = x(D) G^{(1)}(D) = 1 + D^{23}$ and $y^{(2)}(D) = x(D) G^{(2)}(D) = 1 + D^3 + D^6 + D^7 + D^8 + D^9 + D^{10} + D^{21}$ and the weight of this codeword is 10. But for any $x(D)$ such that $x_0 \neq 0$ and degree $[x(D)] \leq 11 = m$, the weight of

the codeword produced by $x(D)$ is at least 11. Hence $d_{\text{FREE}} \neq r_m$. |

Note that $G^{(1)}(D)$ and $G^{(2)}(D)$ in example 5.1 are relatively prime, so the conjecture does not hold even when the generator functions are relatively prime polynomials.

The same conjecture has also been made for systematic codes only. However consider the following counterexample.

Example 5.2 Let $G^{(1)}(D) = 1$, $G^{(2)}(D) = G^{(3)}(D) = 1 + D + D^2 + D^4 + D^6 + D^7 + D^8$ for an $R = 1/3$ fixed systematic code with $m=8$. For $x(D) = 1 + D + D^3 + D^6 + D^8 + D^9$, $y^{(1)}(D) = x(D) G^{(1)}(D) = 1 + D + D^3 + D^6 + D^8 + D^9$, $y^{(2)}(D) = y^{(3)}(D) = x(D) G^{(2)}(D) = x(D) G^{(3)}(D) = 1 + D^{17}$, and the weight of this codeword is 10. But for any $x(D)$ such that $x_0 \neq 0$ and degree $[x(D)] \leq 8 = m$, the weight of the codeword produced by $x(D)$ is at least 11. Hence $d_{\text{FREE}} \neq r_m$. |

It is interesting to note, however, that no counterexamples to this conjecture have yet been found for $R = 1/2$ systematic codes. The author has been able to find some codes for which an information sequence with degree greater than m produces a codeword with weight equal to r_m , but none with weight less than r_m . The difficulty may lie in the fact that very long codes are needed to provide counterexamples, and distances are very difficult to calculate for long codes. Also, no counterexamples have been found for systematic codes of rate other than 1/2 whose generator functions are relatively prime polynomials.

The difficulty in finding counterexamples to this conjecture leads one to believe that the true bound is very close to r_m , perhaps r_{2m} , at least for codes whose generator functions are relatively prime polynomials. Hence it is also likely that r_m is always very close to the actual free distance.

B. Calculating d_{FREE}

Unfortunately, there can be no simple, general method of calculating d_{FREE} until a tight bound is obtained on the length of information sequence needed to produce the minimum free weight codeword. Then d_{FREE} can be calculated simply by computing the minimum row distance over the bounded length of information sequence. However there are many tricks which can be used to find or to closely approximate d_{FREE} .

Since $d_i \leq d_{\text{FREE}} \leq r_i$ for all i , d_i and r_i can be successively computed. If at some point j , $d_j = r_j$, then $d_{\text{FREE}} = d_j = r_j$. Property F4 showed that $d_{\text{FREE}} = d_j = r_{j-m}$ for some finite j if the encoder has a feedforward inverse.

Sometimes the free distance of the reverse code, i.e., the code whose generator matrix is described in definition 2.15, is known. If so, the free distance of the original code is the same as that of the reverse code. This follows from the fact that each codeword in the reverse code is the reciprocal of the codeword in the original code produced by the reciprocal information sequence, where the reciprocal

of the sequence $\underline{z} = [z_0, z_1, z_2, \dots]$ is taken to be $[\dots, z_2, z_1, z_0]$. Since the weight of a sequence and its reciprocal are the same, the set of codeword weights of a code and its reverse code are the same. Note that this is true only for d_{FREE} , and not for d_{FD} or d_{DD} .

Very good approximations to d_{FREE} can be found by computing r_i or d_i for as large an i as feasible. For instance, a computer program has been written for use on the Univac 1107 computer at the University Computer Center which calculates d_{61} for $R = 1/2$ fixed codes in just a few minutes. This usually provides a very good approximation to d_{FREE} for codes with encoding memory less than about 50. d_{61} is a lower bound on d_{FREE} and a reasonable upper bound is usually known from the weight of the generator or some short low weight codeword. Hence d_{61} is often known to be exactly d_{FREE} , and if not it is easy to make a close approximation. The values of and bounds on d_{FREE} given in Appendix A were arrived at in this manner. However, for other rates and longer codes, good approximations to d_{FREE} become harder to make.

Example 5.3 Consider the $m = 71$, $R = 1/2$ fixed systematic code whose generator sequence $[g_{0,1}^{(2)}, g_{1,1}^{(2)}, \dots, g_{71,1}^{(2)}] = [g_0, g_1, \dots, g_{71}]$ is represented three digits at a time in octal notation, starting with g_0 , as $[651, 102, 104, 121, 022, 041, 101, 101]$. This code is known to have $d_{\text{FD}} = 21$. But since the weight of the generator is only 21, $d_{\text{FREE}} \leq 21$. Therefore $d_{\text{FREE}} = d_{\text{FD}} = 21$. This code is one which

will be constructed in Chapter VI. |

Example 5.4 Consider the following $m = 35$, $R = 1/2$ fixed systematic code with $[g_0, g_1, g_2, \dots, g_{35}] = [715, 473, 701, 317]$. This code has $d_{61} = 18$. But the codeword whose transform is $(1+D) G(D)$ has weight 18. Therefore $d_{\text{FREE}} = 18$. This code was constructed by Forney [28] and is presently being used by NASA in its Pioneer satellite series. |

Example 5.5 The $m = 35$, $R = 1/2$ fixed systematic code with $[g_0, g_1, \dots, g_{35}] = [653, 134, 307, 713]$ has $d_{61} = 19$. Since the generator has weight 22, $19 \leq d_{\text{FREE}} \leq 22$. This code is due to Lin and Lyne [29]. |

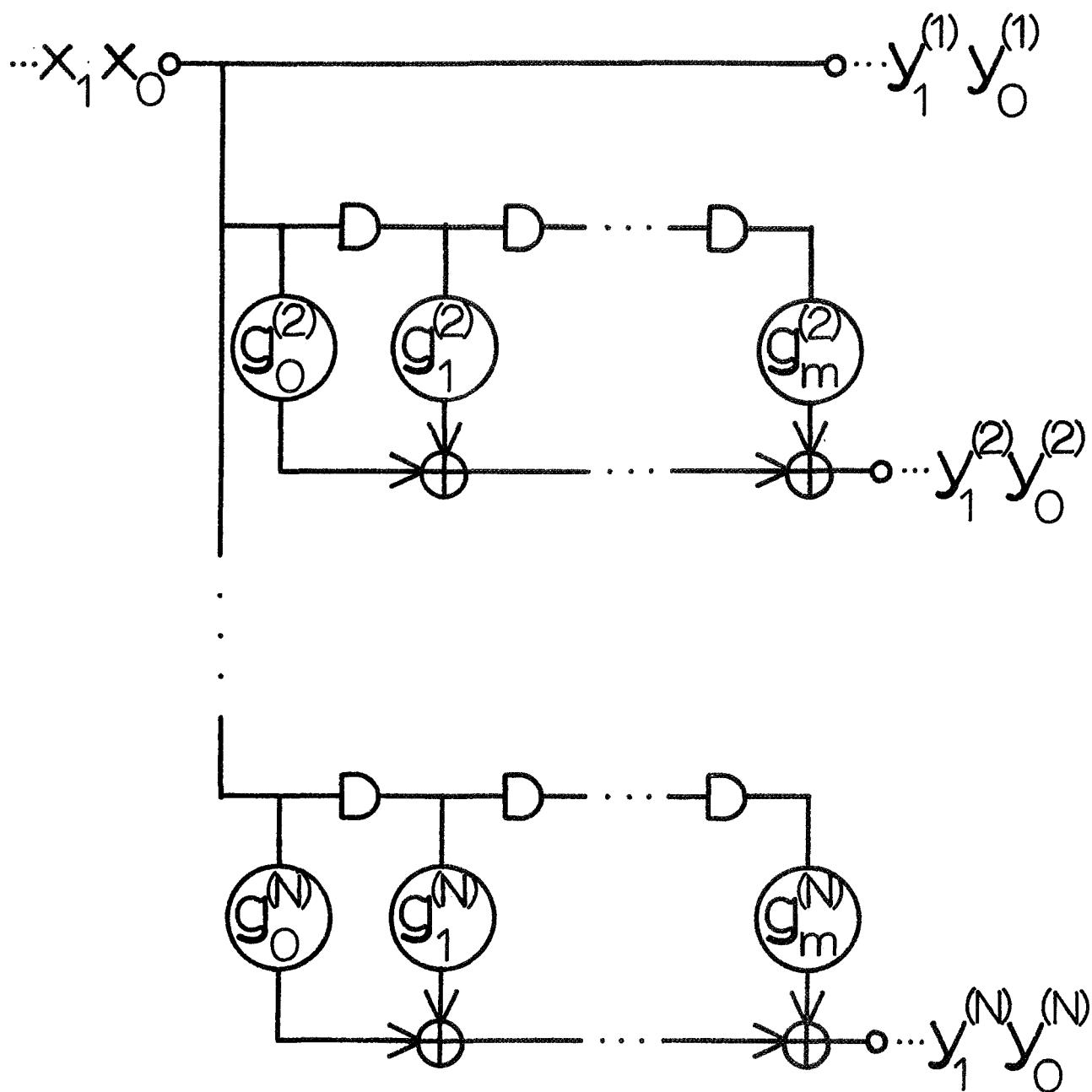
VI. Constructing Good Convolutional Codes

A. The Minimum Weight Construction Algorithms

In this chapter various algorithms for constructing $R = \frac{1}{N}$ fixed binary convolutional codes will be given. In constructing good codes it is desirable that the complexity of the encoder be kept as small as possible. For example, if a code is being used to communicate information from a space vehicle to earth, the encoder is a hardware device on the space vehicle itself. The usual encoding circuit for an $R = \frac{1}{N}$ canonic systematic fixed binary convolutional code is shown in Figure 6.1. Note that the number of two-input modulo-two adders required to implement this encoder is exactly $w_H(\underline{q}) - N$, where \underline{q} is the generator. Therefore minimizing $w_H(\underline{q})$ for a given distance and constraint length minimizes the number of modulo-two adders in the encoder realization. All the codes presented in Section VI.A will exhibit this property, i.e., for a given distance and constraint length, $w_H(\underline{q})$ will have its minimum possible value. As in Chapter II, $[\underline{q}]_j$ will be used to denote the first $(j + 1)N$ entries in \underline{q} .

1. An Algorithm for Finding Good $R = 1/2$ Convolutional Codes

In this section, a simple algorithm will be given which will be shown to produce good $R = 1/2$ canonic systematic fixed binary convolutional codes for all $m \leq 71$. First a statement of the algorithm is given and then several interesting properties of the codes produced are shown. (For



$\text{g}_i^{(j)}$ = multiplication in $\text{GF}(2)$ by $g_i^{(j)}$.

Fig. 6.1. An $R = \frac{1}{N}$ canonic systematic fixed encoder.

convenience, let $g_{j1}^{(2)} = g_j$.)

Algorithm A1

- (0) Set $g_0 = 1$, $d_0 = 2$, and $j = 1$.
- (1) Set $g_j = 1$.
- (2) Compute d_j . If $d_j > d_{j-1}$, go to (4).
- (3) Set $g_j = 0$.
- (4) If $j = m$, stop. Otherwise, set $j = j+1$ and go to (1).

Property A1-1 $w_H([g]_j) = d_j$ for $j = 0, 1, \dots, m$.

Proof $w_H([g]_j) \geq d_j$ by property C3 of the column distance. However, since $w_H(g_0) = d_0 = 2$ and since g_j is permanently set to 1, i.e., $w_H([g]_j)$ is increased by one, if and only if $d_j > d_{j-1}$, $w_H([g]_j) \leq d_j$. Therefore $w_H([g]_j) = d_j$. |

Since properties C1 and C3 of the column distance require that $w_H(g) = w_H([g]_m) \geq d_m = d_{FD}$, property A1-1 ensures that $w_H(g)$ is minimal and hence the resultant code requires the minimum number of modulo-two adders in its encoding circuit.

Property A1-2 If $g_j = 1$, then $g_{j+1} = 0$, for every $j \neq 0$.

Proof Assume $g_j = 1$, $j \geq 1$ (note that algorithm A1 sets $g_0 = g_1 = 1$). Then set $g_{j+1} = 1$. The information sequence

$$[\underline{x}]_{j+1} = \left[x_0^{(1)} \ x_1^{(1)} \ \dots \ x_{j-1}^{(1)} \ x_j^{(1)} \ x_{j+1}^{(1)} \right] = [1, 0, \dots, 0, 1, 0]$$

always produces a codeword with $d_{j+1} = d_j$. Therefore algorithm A1 will set $g_{j+1} = 0$. |

Property Al-2 allows us automatically to add a 0 to $[g]_j$ after adding each 1 beyond g_0 . This permits a shortcut to reduce the number of times steps 1 and 2 must be applied to reach a given length code.

Property Al-3 Let g be the generator obtained by using algorithm Al. Let $g' \neq g$ be the generator of any other $R = 1/2$ canonic systematic fixed code of the same length such that $w_H([g'])_j = d'_j$, $j = 0, 1, \dots, m$, i.e., such that each 1 in the generator increases the column distance by one. Then there exists a j_0 , $0 \leq j_0 \leq m$, such that $d_{j_0} > d'_{j_0}$ and $d_i = d'_i$, $i = 0, 1, 2, \dots, j_0 - 1$.

Proof Assume the first point at which the two generators disagree, j_0 , $0 \leq j_0 \leq m$, has $g_{j_0} = 0$, $g'_{j_0} = 1$. Then $d'_{j_0} = d_{j_0} + 1 > d_{j_0}$. But this is impossible, since if the column distance can increase at j_0 algorithm Al would make $g_j = 1$. Therefore, the first point at which the two generators disagree must have $g'_{j_0} = 0$, $g_{j_0} = 1$, and hence, $d_{j_0} > d'_{j_0}$. |

Property Al-3 shows that any other algorithm for generating $R = 1/2$ fixed canonic systematic convolutional codes which increases the column distance by one each time a 1 is added to the generator differs from algorithm Al in that such 1's are not always added at the first opportunity. Note also that in the computation of step (2), if $d_j > d_{j-1}$, then $d_j = d_{j-1} + 1$, and if g_j is set to 0 in step (3), then $d_j = d_{j-1}$; and that the codes obtained from algorithm Al exhibit the

"nested" property, i.e., for $m_1 < m_2$, $[g]_{m_2} = \left[[g]_{m_1}, 0, g_{m_1+1}, \dots, 0, g_{m_2} \right]$.

Algorithm A1 was programmed on the Univac 1107 computer at the University Computer Center. The most difficult part of algorithm A1 to program is the computation of d_j in step (2). This was done by using a sequential-decoding-like algorithm suggested by Forney [28]. The flow chart for this algorithm, called SEAL, is shown in Figure 6.2. The flow chart for algorithm A1 is then shown in Figure 6.3.

The codes obtained from algorithm A1 are compared with Bussgang's [30] optimal codes and Lin and Lyne's [29] near-optimal codes in Table 6.1. Bussgang's computer search for optimal codes reached $m = 15$ before the amount of computation became too large. Lin and Lyne carried their near-optimal search out to $m = 20$ (Forney [28] has extended this to $m = 48$). Algorithm A1 is sufficiently simple to allow hand computation out to $m = 22$ and it was extended to $m = 71$ by computer. Table 6.1 also compares the codes obtained with the non-asymptotic Gilbert lower bound [30], and it can be seen that the codes remain good out to $m = 71$. The adjoints of the codes obtained from algorithm A1, which are known to have exactly the same set of codeword weights over the first constraint length [30], are also given in Table 6.1.

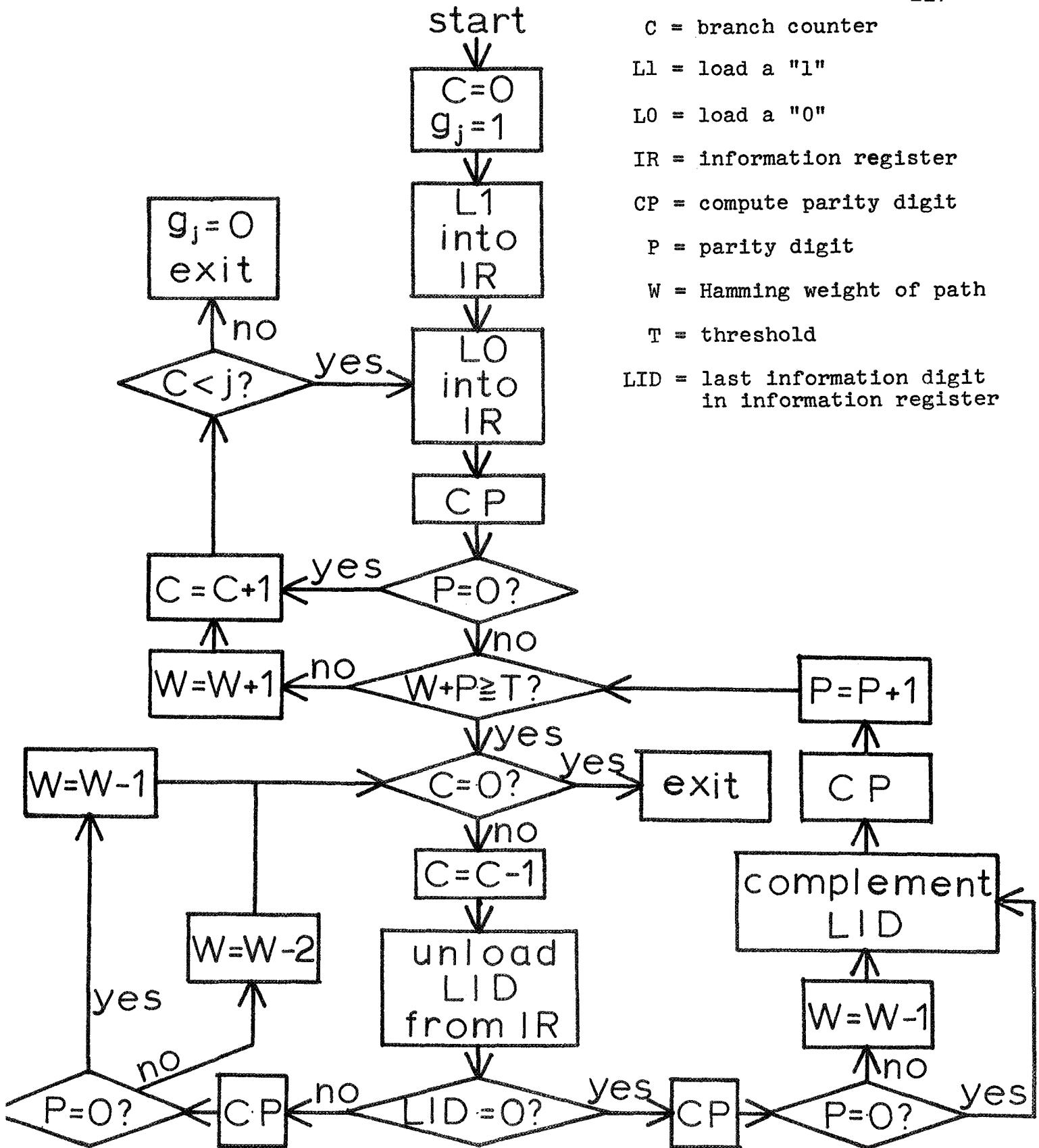


Fig. 6.2. SEAL flow chart.

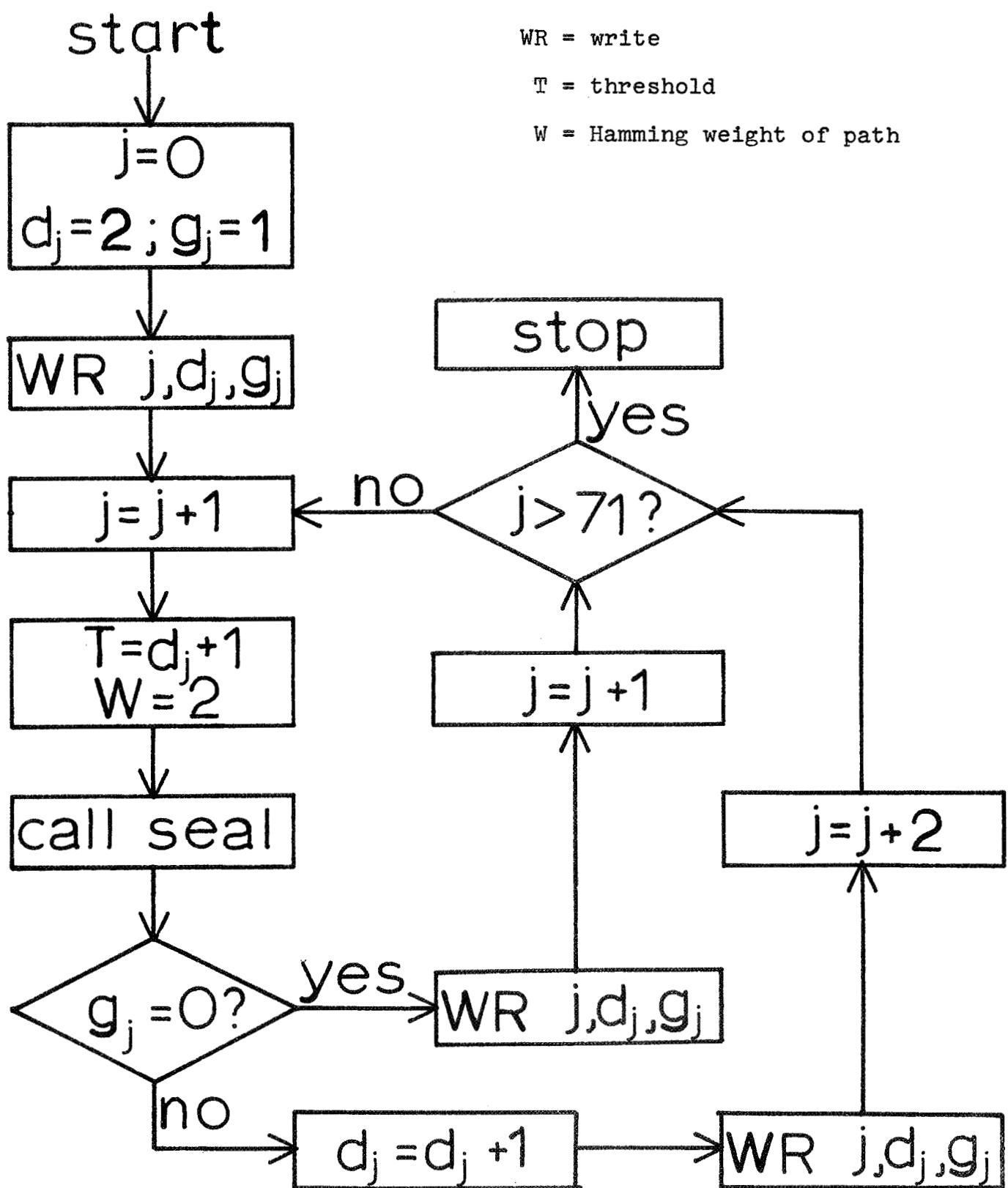


Fig. 6.3. Flow chart for algorithm A1.

TABLE 6.1
COMPARISON OF R = 1/2 CODES

j	g_j	\hat{g}_j	d_j	d_G	d_{LL}	d_B	j	g_j	\hat{g}_j	d_j	d_G	d_{LL}
0	1	1	2	2	2	2	36	0	0	13	11	14
1	1	1	3	3	3	3	37	0	1	13	12	14
2	0	1	3	3	3	3	38	0	0	13	12	14
3	1	0	4	4	4	4	39	0	1	13	12	15
4	0	1	4	4	4	4	40	1	0	14	12	15
5	1	1	5	4	5	5	41	0	0	14	13	15
6	0	0	5	4	5	5	42	0	0	14	13	15
7	0	0	5	5	5	6	43	1	0	15	13	16
8	1	0	6	5	6	6	44	0	0	15	13	16
9	0	0	6	5	6	6	45	0	1	15	13	16
10	0	0	6	5	7	7	46	0	0	15	14	16
11	1	1	7	6	7	7	47	0	1	15	14	17
12	0	1	7	6	7	8	48	1	1	16	14	17
13	0	1	7	6	8	8	49	0	1	16	14	
14	0	0	7	6	8	8	50	0	0	16	15	
15	0	0	7	7	9	9	51	0	1	16	15	
16	1	0	8	7	9		52	0	0	16	15	
17	0	0	8	7	9		53	1	0	17	15	
18	0	0	8	7	9		54	0	0	17	15	
19	0	1	8	8	10		55	0	1	17	16	
20	1	0	9	8	10		56	1	0	18	16	
21	0	1	9	8	10		57	0	1	18	16	
22	0	0	9	8	10		58	0	1	18	16	
23	0	1	9	9	10		59	0	0	18	16	
24	1	0	10	9	11		60	0	0	18	17	
25	0	0	10	9	11		61	0	1	18	17	
26	0	1	10	9	11		62	1	0	19	17	
27	1	0	11	9	11		63	0	0	19	17	
28	0	0	11	10	12		64	0	1	19	18	
29	0	1	11	10	12		65	1	0	20	18	
30	0	0	11	10	12		66	0	0	20	18	
31	1	1	12	10	12		67	0	1	20	18	
32	0	0	12	11	13		68	0	1	20	18	
33	0	0	12	11	13		69	0	1	20	19	
34	0	0	12	11	13		70	0	0	20	19	
35	1	0	13	11	14		71	1	1	21	19	

$d_G = d_{\text{GILBERT BOUND}}$ $d_{LL} = d_{\text{LIN AND LYNE}}$ $d_B = d_{\text{BUSSGANG}}$

\hat{g}_j = adjoint codes

An interesting, but as yet unsolved, question is whether algorithm A1 will continue to produce good codes, i.e., codes whose column distance increases linearly with j , as j becomes arbitrarily large. The amount of computation required by algorithm A1, because of the calculation of d_j in step (2), appears to increase exponentially with increasing j , as it does in all known search techniques for finding codes. However, because of its simplicity, algorithm A1 requires less computation than other known search techniques.

2. Algorithms for Generating Good $R = \frac{1}{3}$ and $R = \frac{1}{4}$ Codes

For rates $R = \frac{1}{N}$, $N > 2$, an algorithm is sought for generating codes such that $d_j = w_H([g]_j)$, $j = 0, 1, 2, \dots, m$, and 1's are added to the generator at the first opportunity consistent with this constraint. Since there are now $N - 1$ digits, viz. $g_{j1}^{(2)}, g_{j1}^{(3)}, \dots, g_{j1}^{(N)}$, to be specified in each block, there will not be a unique algorithm with the above property for $N > 2$. For example, for $N = 3$ the three following algorithms each result in a code such that $d_j = w_H([g]_j)$ and "ones" are added to the generator at the earliest opportunity. For $N = 3$, it is well known [29] that $d_j \leq d_{j-1} + 1$ so that it is unnecessary to test the choice $g_{j1}^{(2)} = g_{j1}^{(3)} = 1$ since the column distance can never increase by 2. (For convenience let $g_{j1}^{(2)} = g_j^{(2)}$ and $g_{j1}^{(3)} = g_j^{(3)}$.)

Algorithm A2

- (0) Set $g_0^{(2)} = g_0^{(3)} = 1$, $d_0 = 3$, and $j = 1$.
- (1) Set $g_j^{(2)} = 1$, $g_j^{(3)} = 0$.
- (2) Compute d_j . If $d_j > d_{j-1}$, go to (6).
- (3) Set $g_j^{(2)} = 0$, $g_j^{(3)} = 1$.
- (4) Compute d_j . If $d_j > d_{j-1}$, go to (6).
- (5) Set $g_j^{(2)} = g_j^{(3)} = 0$.
- (6) If $j = m$, stop. Otherwise, set $j = j + 1$ and go to (1).

Algorithm A3

Steps (0) through (5) are the same as in algorithm A2.

- (6) If $j = m$, stop. Otherwise, interchange steps (1) and (3), set $j = j + 1$, and go to (1).

Algorithm A4

Steps (0) through (5) are the same as in algorithm A2.

- (6) If $j = m$, stop. Otherwise, if d_j increased during step (2), interchange steps (1) and (3), set $j = j + 1$, and go to (1). If d_j increased during step (4) or remained the same, set $j = j + 1$ and go to (1).

The codes obtained from algorithms A2, A3, and A4 are shown in Table 6.2 and are compared to Bussgang's codes, Lin and Lyne's codes, and to the non-asymptotic Gilbert lower bound. Each algorithm was carried out to $m = 35$ by computer.

TABLE 6.2
COMPARISON OF R = 1/3 CODES

j				Algorithm A2			Algorithm A3			Algorithm A4		
	d_G	d_{LL}	d_B	$g_j^{(2)}$	$g_j^{(3)}$	d_j	$g_j^{(2)}$	$g_j^{(3)}$	d_j	$g_j^{(2)}$	$g_j^{(3)}$	d_j
0	3	3	3	1	1	3	1	1	3	1	1	3
1	4	4	4	1	0	4	1	0	4	1	0	4
2	5	5	5	1	0	5	0	1	5	0	1	5
3	6	6	6	0	1	6	1	0	6	1	0	6
4	6	7	7	1	0	7	1	0	7	1	0	7
5	7	8	8	0	1	8	1	0	8	1	0	8
6	8	9	9	0	1	9	0	0	8	0	0	8
7	8	9		0	1	10	1	0	9	0	1	9
8	9	10		0	0	10	0	1	10	1	0	10
9	9	11		1	0	11	0	0	10	0	0	10
10	10	12		1	0	12	1	0	11	0	1	11
11	10	12		0	0	12	0	1	12	0	1	12
12	11	13		1	0	13	0	0	12	1	0	13
13	11	14		0	0	13	0	0	12	0	0	13
14	12	15		1	0	14	0	1	13	1	0	14
15	12	15		1	0	15	1	0	14	0	0	14
16	13	16		0	0	15	1	0	15	0	1	15
17	14	16		1	0	16	0	0	15	0	0	15
18				0	1	17	0	1	16	1	0	16
19				0	0	17	1	0	17	1	0	17
20				1	0	18	0	0	17	1	0	18
21				0	0	18	0	1	18	1	0	19
22				1	0	19	0	0	18	0	0	19
23				0	1	20	1	0	19	0	0	19
24				0	0	20	0	0	19	0	1	20
25				0	0	20	1	0	20	1	0	21
26				1	0	21	0	1	21	0	0	21
27				1	0	22	0	0	21	0	0	21
28				0	1	23	0	1	22	0	1	22
29				0	0	23	0	1	23	0	0	22
30				0	0	23	0	0	23	1	0	23
31				0	1	24	0	0	23	1	0	24
32				0	0	24	1	0	24	0	0	24
33				1	0	25	1	0	25	1	0	25
34				0	0	25	0	0	25	0	1	25
35				1	0	26	1	0	26	0	0	26

 $d_G = d_{\text{GILBERT BOUND}}$ $d_{LL} = d_{\text{LIN AND LYNE}}$ $d_B = d_{\text{BUSSGANG}}$

Again the codes are quite good and are considerably longer than other known good $R = 1/3$ codes. Note that the codes obtained from Algorithms A2, A3, and A4 exhibit about the same distance properties. Indeed it seems the many variations of the algorithm available for $R = 1/3$ will have little effect on the distance properties of the resulting codes. The two adjoints of each of the codes obtained from Algorithms A2, A3, and A4 are given in Table 6.3.

Note that at $m = 7$, the code obtained from Algorithm A2 has greater feedback decoding minimum distance than Lin and Lyne's near-optimal code. It can be shown that this code meets the Plotkin upper bound [8] on feedback decoding minimum distance at $m = 7$.

To generate $R = 1/4$ codes, $g_{jl}^{(2)}$, $g_{jl}^{(3)}$, and $g_{jl}^{(4)}$ must be specified for each j , and it must be recognized that an increase of either one or two in the column distance for each j is possible. Only one algorithm will be given for generating $R = 1/4$ codes with the property that $d_j = w_H$ ($[g_j]$) and 1's are added to the generator at the earliest opportunity. (For convenience, let $g_{j1}^{(2)} = g_j^{(2)}$, $g_{j1}^{(3)} = g_j^{(3)}$, and $g_{j1}^{(4)} = g_j^{(4)}$.)

Algorithm A5

- (0) Set $g_0^{(2)} = g_0^{(3)} = g_0^{(4)} = 1$, $d_0 = 4$, and $j = 1$.
- (1) Set $g_j^{(2)} = g_j^{(3)} = 1$, $g_j^{(4)} = 0$, $i = 1$, and go to (8).

- (2) Set $g_j^{(3)} = 0$, $g_j^{(4)} = 1$, $i = 2$, and go to (8).
- (3) Set $g_j^{(2)} = 0$, $g_j^{(3)} = 1$, $i = 3$, and go to (8).
- (4) Set $g_j^{(3)} = 0$, $i = 4$, and go to (8).
- (5) Set $g_j^{(3)} = 1$, $g_j^{(4)} = 0$, $i = 5$, and go to (8).
- (6) Set $g_j^{(2)} = 1$, $g_j^{(3)} = 0$, $i = 6$, and go to (8).
- (7) Set $g_j^{(2)} = 0$ and go to (9).
- (8) Compute d_j . If $d_j = d_{j-1}$, go to (i + 1).
- (9) If $j = m$, stop. Otherwise, set $j = j + 1$ and go to (1).

Table 6.4 compares the $R = 1/4$ codes generated by Algorithm A5, Lin and Lyne's codes, and the non-asymptotic Gilbert lower bound. Algorithm A5 was carried out to $m = 35$ by computer and again good codes were found. The three adjoints of the code produced by Algorithm A5 are given in Table 6.5. Clearly properties Al-1 and Al-3, as well as the "nested" property, also hold for the codes of Algorithms A2 through A5.

3. An Algorithm for Generating $R = 1/2$ Codes with Large Free Distance

Clearly, it is of considerable interest to find codes with known d_{FREE} , especially codes for which $d_{\text{FREE}} > d_{\text{FD}}$. A slight modification of the preceding algorithms can be used for this purpose. Algorithm A6 indicates the necessary modification of Algorithm Al.

TABLE 6.3
ADJOINTS OF R = 1/3 CODES

j	Algorithm A2				Algorithm A3				Algorithm A4			
	1st adj.	2nd adj.	1st adj.	2nd adj.	1st adj.	2nd adj.	1st adj.	2nd adj.	1st adj.	2nd adj.	1st adj.	2nd adj.
	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(2)}$	$g_j^{(3)}$
0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1	0
2	0	0	1	0	1	0	1	1	1	0	1	1
3	1	0	1	1	0	1	0	0	0	1	0	0
4	0	1	0	0	0	1	0	1	0	1	0	1
5	0	1	0	1	1	1	1	0	1	1	1	0
6	0	1	1	0	1	1	0	1	1	1	0	1
7	1	1	1	1	1	0	0	0	0	0	0	1
8	1	1	1	0	1	1	1	0	0	0	0	1
9	1	0	0	1	0	0	1	0	1	0	0	1
10	0	0	1	1	0	0	0	1	1	0	1	0
11	1	0	0	1	0	1	0	1	1	0	1	0
12	0	0	1	1	0	1	0	0	1	0	1	1
13	0	0	0	0	0	0	0	0	0	0	1	0
14	1	1	1	1	1	1	1	0	1	0	1	0
15	1	1	0	1	0	0	0	1	0	1	1	0
16	0	0	1	1	1	1	1	1	1	1	0	1
17	1	1	0	0	1	0	1	0	1	1	1	0
18	0	1	0	0	0	1	1	1	0	1	1	0
19	0	0	0	1	1	1	1	1	0	0	1	0
20	0	1	1	0	1	1	1	1	1	0	1	1
21	1	0	1	0	1	0	1	1	1	1	0	0
22	0	0	0	1	1	1	1	0	1	1	1	0
23	1	1	1	1	1	1	0	1	1	0	1	0
24	1	0	1	1	1	1	0	0	1	0	0	0
25	1	0	1	1	1	1	1	0	1	0	1	0
26	1	1	0	1	1	0	0	0	1	1	1	0
27	0	1	0	1	1	0	0	0	0	1	1	1
28	0	0	1	0	0	0	0	0	0	1	1	0
29	1	0	0	1	0	1	0	1	0	0	0	0
30	0	0	0	0	1	1	0	1	1	1	0	1
31	0	0	0	1	1	0	0	1	1	0	0	1
32	1	1	1	0	1	0	1	1	1	0	0	1
33	0	0	0	1	1	1	0	0	1	1	1	0
34	0	1	0	0	0	0	0	0	1	1	0	1
35	0	0	1	1	0	0	0	1	0	1	1	0

TABLE 6.4
COMPARISON OF $R = 1/4$ CODES

j	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(4)}$	d_j	d_G	d_{LL}
0	1	1	1	4	4	4
1	1	1	0	6	6	6
2	1	0	1	8	7	8
3	0	0	1	9	8	9
4	0	1	0	10	9	10
5	0	0	1	11	10	11
6	0	0	1	12	11	13
7	1	0	1	14	12	14
8	0	0	1	15	13	15
9	0	1	0	16	13	16
10	0	1	0	17	14	17
11	0	0	0	17	15	18
12	1	1	0	19	16	19
13	0	0	1	20		21
14	0	1	0	21		22
15	0	0	1	22		23
16	0	0	1	23		
17	0	0	1	24		
18	0	1	0	25		
19	0	1	0	26		
20	1	0	0	27		
21	0	0	1	28		
22	0	0	1	29		
23	0	0	1	30		
24	0	1	0	31		
25	0	0	1	32		
26	0	1	0	33		
27	0	1	0	34		
28	0	0	0	34		
29	0	0	1	35		
30	0	1	0	36		
31	0	1	0	37		
32	0	0	0	37		
33	1	1	0	39		
34	0	0	0	39		
35	0	0	1	40		

$d_G = d_{\text{GILBERT BOUND}}$

$d_{LL} = d_{\text{LIN AND LYNE}}$

TABLE 6.5
ADJOINTS OF R = 1/4 CODES

j	1st adjoint			2nd adjoint			3rd adjoint		
	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(4)}$	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(4)}$	$g_j^{(2)}$	$g_j^{(3)}$	$g_j^{(4)}$
0	1	1	1	1	1	1	1	1	1
1	1	0	1	0	1	1	1	1	0
2	0	1	1	1	1	0	0	1	1
3	1	1	1	1	1	1	0	0	1
4	1	1	0	0	0	0	1	1	1
5	0	0	0	0	1	0	1	0	1
6	1	1	1	1	0	1	1	1	1
7	0	0	1	1	1	1	1	0	0
8	0	1	0	1	1	0	0	0	1
9	0	1	0	0	1	1	0	1	0
10	1	0	1	0	1	0	1	1	1
11	0	0	1	0	0	0	1	1	1
12	0	0	0	1	0	0	0	1	0
13	0	0	1	1	0	1	0	0	0
14	0	1	1	0	0	1	0	0	1
15	1	0	1	0	0	0	1	1	0
16	0	1	0	0	0	1	0	1	0
17	0	1	1	1	0	1	0	0	1
18	1	0	1	0	1	1	1	0	1
19	1	0	0	0	1	0	1	0	0
20	1	1	1	1	1	0	0	1	1
21	1	1	1	1	0	1	1	1	1
22	0	1	1	0	0	1	0	1	1
23	0	0	0	0	0	1	1	0	1
24	1	0	0	0	1	0	1	1	1
25	0	1	0	1	0	1	1	0	1
26	1	0	1	0	0	1	1	1	1
27	1	1	0	0	0	1	1	0	1
28	1	1	0	1	1	0	1	1	1
29	0	0	0	1	0	1	1	1	1
30	1	1	0	1	0	1	1	1	1
31	1	1	0	0	1	0	0	1	0
32	1	1	1	0	0	1	1	0	1
33	1	1	1	0	1	0	0	1	0
34	0	1	1	0	0	1	0	1	0
35	1	1	1	0	0	0	0	0	1

Algorithm A6 (Assume $L \geq m$.)

- (0) Set $g_0 = 1$, $D_0 = 2$, and $j = 1$.
- (1) Set $g_j = 1$.
- (2) Compute d_L . If $d_L > D_{j-1}$, set $D_j = d_L$ and go to (4).
- (3) Set $g_j = 0$ and $D_j = D_{j-1}$.
- (4) If $j = m$, stop. Otherwise, set $j = j + 1$ and go to (1).

The following properties of the codes resulting from Algorithm A6 will be presented without proof, since the proofs are similar to those used to prove the properties of Algorithm A1.

Property A6-1 $w_H([g]_j) = D_j$ for all j .

Property A6-2 In the computation of step (2), if $d_L > D_{j-1}$, then $D_j = D_{j-1} + 1$.

Property A6-3 The codes obtained from Algorithm A6 exhibit the "nested" property.

Theorem 6.1 $w_H([g]_j) = D_j = d_{\text{FREE}}$ for all j , where d_{FREE} is the free distance of the code with memory order j .

Proof $w_H([g]_j) = D_j \leq d_{\text{FREE}}$ by property A6-1 and theorem 2.4. $d_{\text{FREE}} \leq w_H([g]_j)$ by property F2 of the free distance.

Therefore $d_{\text{FREE}} = w_H([g]_j)$ for all j .

Theorem 6.2 For all the codes obtained from Algorithms A1, A2, A3, A4, and A5, $d_{\text{FREE}} = d_{\text{FD}} = d_m$.

Proof $d_{\text{FD}} = d_m = w_H([g]_m)$ is a property of the codes obtained from Algorithms A1, A2, A3, A4, and A5, and $w_H([g]_m)$

$= d_{\text{FREE}}$ follows from property F2. |

In general Algorithm A6 will result in generators with greater weight than those obtained from Algorithm A1. Therefore, d_{FREE} for the codes obtained from Algorithm A6 will be larger than d_{FD} for the same length codes obtained from Algorithm A1. Clearly, it is wise to choose L as large as is computationally possible in Algorithm A6.

Table 6.6 shows the results of applying Algorithm A6 to the construction of an $R = 1/2$ fixed canonic systematic binary code with $m = 35$ and $L = 71$. The adjoint of this code has $d_{\text{FREE}} = 18$ and is also given in Table 6.6. It is interesting to note that Algorithm A1 produced a code with $m = 35$ and $d_{\text{FREE}} = d_{\text{FD}} = 13$. Algorithm A6 resulted in a code with $m = 35$ and $d_{\text{FREE}} = 17$. d_{FD} was checked for this code and found to be 13. Therefore, Algorithm A6 produced a code with the same length and the same d_{FD} , but with a larger d_{FREE} . Although the two codes have the same d_{FD} , the code obtained from Algorithm A6 should exhibit a lower probability of error when used with sequential decoding. This was verified by simulating a sequential decoder for use on a binary symmetric channel on the Univac 1107. The results of this simulation will be presented in Section VI.C.

Note that all the codes constructed in Section VI.A have distances considerably better than the non-asymptotic Gilbert lower bound and are longer than any previously known good codes. Also, each code has the property of minimizing the

TABLE 6.6
CODES OBTAINED FROM ALGORITHM A6

j	g_j	d_{FREE}	\hat{g}_j
0	1	2	1
1	1	3	1
2	1	4	0
3	0	4	1
4	1	5	0
5	1	6	1
6	0	6	0
7	1	7	1
8	0	7	1
9	1	8	0
10	0	8	0
11	0	8	1
12	1	9	0
13	1	10	0
14	0	10	1
15	0	10	0
16	0	10	0
17	0	10	0
18	1	11	0
19	1	12	0
20	1	13	1
21	0	13	1
22	0	13	1
23	0	13	0
24	0	13	0
25	1	14	1
26	1	15	0
27	1	16	0
28	0	16	0
29	0	16	1
30	0	16	0
31	0	16	0
32	0	16	1
33	0	16	1
34	0	16	1
35	1	17	1

\hat{g}_j = adjoint codes

NOTE: The adjoint code with $m = 35$ has $d_{\text{FREE}} = 18$.

number of modulo-2 adders needed in the encoding circuit for codes of a given distance and constraint length.

B. More Construction Algorithms for $R = 1/2$ Codes

It was shown in theorem 6.2 that the codes produced by Algorithms A1 through A5 have $d_{\text{FREE}} = d_{\text{FD}}$. This is a direct result of minimizing the number of modulo-2 adders needed in the encoding circuit. Hence this property must be abandoned if codes with d_{FREE} considerably larger than d_{FD} are to be obtained. In this section a number of algorithms are presented for producing fixed $R = 1/2$ binary codes with large free distance.

1. Systematic $R = 1/2$ Codes

Since a low density of 1's in the generator necessarily produces a code with low d_{FREE} , the following algorithm was designed to produce a high density of 1's in the generator.

(For convenience, let $g_{jl}^{(2)} = g_{j.}$)

Algorithm A7

- (0) Set $g_0 = 1$, $d_0 = 2$, and $j = 1$.
- (1) Set $g_j = 0$.
- (2) Compute d_j . If $d_j > d_{j-1}$, go to (4).
- (3) Set $g_j = 1$ and compute d_j .
- (4) If $j = m$, stop. Otherwise, set $j = j + 1$ and go to (1).

This code and its adjoint are shown for $m = 35$ in Table 6.7 along with the non-asymptotic Gilbert lower bound. Note that

TABLE 6.7
CODES OBTAINED FROM ALGORITHM A7

j	g_j	\hat{g}_j	d_j	d_G
0	1	1	2	2
1	1	1	3	3
2	1	0	3	3
3	0	1	4	4
4	1	0	4	4
5	1	1	5	4
6	1	1	5	4
7	1	1	5	5
8	0	0	6	5
9	1	0	6	5
10	1	1	6	5
11	0	1	7	6
12	1	1	7	6
13	1	1	7	6
14	1	1	8	6
15	1	1	8	7
16	1	0	8	7
17	1	0	9	7
18	1	1	9	7
19	1	0	9	8
20	1	0	9	8
21	1	0	9	8
22	1	1	9	8
23	1	0	9	9
24	0	0	10	9
25	1	1	10	9
26	1	1	10	9
27	1	1	10	9
28	0	1	11	10
29	1	0	11	10
30	1	1	11	10
31	1	0	11	10
32	1	1	12	11
33	1	0	12	11
34	0	0	13	11
35	1	1	13	11

\hat{g}_j = adjoint codes

d_G = $d_{\text{GILBERT BOUND}}$

d_j must be recomputed in step (3) since, unlike Algorithm A1 where only the choice $g_j = 1$ can increase d_j , either $g_j = 0$ or $g_j = 1$ can increase d_j .

Since it is known that a randomly constructed code is with high probability a good code [31], the following algorithm was designed to keep the number of 1's and 0's in the generator sequence $[g_{01}^{(2)}, g_{11}^{(2)}, g_{21}^{(2)}, \dots]$ about equal. (For convenience, let $g_{j1}^{(2)} = g_j$.)

Algorithm A8

- (0) Set $g_0 = 1$, $d_0 = 2$, $w = 0$, and $j = 1$.
- (1) Set $g_j = 0$.
- (2) Compute d_j . If $d_j > d_{j-1}$, go to (7).
- (3) Set $g_j = 1$.
- (4) Compute d_j . If $d_j > d_{j-1}$, set $w = w + 2$ and go to (7).
- (5) If $j > w$, set $w = w + 2$ and go to (7).
- (6) Set $g_j = 0$.
- (7) If $j = m$, stop. Otherwise, set $j = j + 1$ and go to (1).

This code and its adjoint are given for $m = 35$ in Table 6.8 along with the non-asymptotic Gilbert lower bound. Again note that either $g_j = 0$ or $g_j = 1$ can increase d_j .

Algorithm A8 can be modified to provide merely an extension of Bussgang's optimal codes from $m = 15$ to $m = 35$. Both of Bussgang's optimal $m = 15$ codes were extended using Algorithm A8 and the resulting codes, along with their adjoints and the non-asymptotic Gilbert lower bound, are given

TABLE 6.8
CODES OBTAINED FROM ALGORITHM A8

j	g_j	\hat{g}_j	d_j	d_G
0	1	1	2	2
1	1	1	3	3
2	0	1	3	3
3	1	0	4	4
4	0	1	4	4
5	1	1	5	4
6	0	0	5	4
7	1	1	5	5
8	1	0	6	5
9	0	1	6	5
10	0	0	7	5
11	1	0	7	6
12	0	1	7	6
13	1	0	8	6
14	0	1	8	6
15	1	0	8	7
16	0	1	8	7
17	1	1	8	7
18	1	1	9	7
19	0	1	9	8
20	0	1	9	8
21	1	0	9	8
22	0	0	9	8
23	0	0	10	9
24	1	0	10	9
25	1	1	10	9
26	0	1	11	9
27	1	1	11	9
28	0	1	11	10
29	1	0	11	10
30	0	0	11	10
31	0	1	12	10
32	1	1	12	11
33	1	0	12	11
34	0	1	13	11
35	1	0	13	11

\hat{g}_j = adjoint codes

d_G = $d_{\text{GILBERT BOUND}}$

in Table 6.9. Finally d_{FD} and d_{FREE} are given for each of the eight codes constructed in this section in Table 6.10. Again, d_{FD} is the same for a code and its adjoint, but d_{FREE} is not necessarily the same since the set of codeword weights are identical only over one constraint length. For all except the code of Algorithm A7, it is possible to give only a range for d_{FREE} . Note, however, that for each code d_{FREE} is considerably larger than d_{FD} and that d_{FD} is about the same as for the $m = 35$, $R = 1/2$ codes constructed in Section VI.A. Hence it is reasonable to expect that these codes should perform considerably better with sequential decoding than those of the previous section. That they do will be demonstrated in Section VI.C.

2. A Non-systematic $R = 1/2$ Code

It has been noted that non-systematic codes are capable of providing a lower probability of decoding error than systematic codes for sequential decoding. However it is also desirable that the "quick-look" and ease of implementation properties of systematic codes be retained. Massey [32] has shown that the class of fixed binary $R = 1/2$ non-systematic codes for which a Gilbert lower bound on d_{FD} was proved in Section IV.D has these properties. For these codes

$$G_1^{(2)}(D) = D + G_1^{(1)}(D) \text{ or } G^{(2)}(D) = D + G^{(1)}(D). \quad (143)$$

Therefore, for any information sequence whose transform is $x(D)$,

TABLE 6.9
EXTENSIONS OF BUSSGANG'S OPTIMAL M = 15 CODES

j	BUSSGANG 1				BUSSGANG 2		
	d _G	g _j	ĝ _j	d _j	g _j	ĝ _j	d _j
0	2	1	1		1	1	
1	3	1	1		1	1	
2	3	0	1		1	0	
3	4	1	0		0	1	
4	4	0	1		1	0	
5	4	1	1		1	1	
6	4	0	0		0	0	
7	5	1	1		1	1	
8	5	1	0		0	1	
9	5	0	1		1	0	
10	5	0	0		0	0	
11	6	1	0		0	1	
12	6	0	1		1	0	
13	6	1	0		0	1	
14	6	1	0		0	1	
15	7	1	0	9	0	1	9
16	7	0	0	9	1	1	9
17	7	0	0	9	1	1	9
18	7	0	1	9	0	0	9
19	8	1	1	9	0	1	10
20	8	1	1	10	1	0	10
21	8	0	1	10	1	1	10
22	8	0	1	10	0	0	10
23	9	1	1	10	1	0	10
24	9	0	1	11	0	1	10
25	9	1	0	11	0	0	11
26	9	0	1	11	1	0	11
27	9	1	0	11	1	1	11
28	10	0	0	12	1	1	12
29	10	1	1	12	0	0	12
30	10	0	1	12	0	1	12
31	10	0	1	13	1	1	12
32	11	1	1	13	0	1	12
33	11	1	0	13	0	0	13
34	11	0	0	13	1	1	13
35	11	1	1	13	1	1	13

ĝ_j = adjoint codes

d_G = d_{GILBERT BOUND}

TABLE 6.10
DISTANCE PROPERTIES OF $8 \times m = 35$ CODES

CODE	d_{FD}	d_{FREE}
Algorithm A7	13	16
A7 adjoint	13	$18 \leq d_{FREE} \leq 22$
Algorithm A8	13	$16 \leq d_{FREE} \leq 20$
A8 adjoint	13	$18 \leq d_{FREE} \leq 22$
A8-BUSSGANG 1	13	$16 \leq d_{FREE} \leq 20$
A8-B1 adjoint	13	$18 \leq d_{FREE} \leq 20$
A8-BUSSGANG 2	13	$18 \leq d_{FREE} \leq 20$
A8-B2 adjoint	13	$17 \leq d_{FREE} \leq 23$

$$\begin{aligned} y^{(1)}(D) + y^{(2)}(D) &= x(D) G^{(1)}(D) + x(D) G^{(2)}(D) = x(D) \\ \left[G^{(1)}(D) + D + G^{(1)}(D) \right] &= D x(D) \end{aligned} \quad (144)$$

and the information sequence can be obtained from the two encoded sequences with a delay of one time unit simply by adding together $y^{(1)}(D)$ and $y^{(2)}(D)$. This allows a quick look at the data sequence to be made before submitting the received sequence to error correction, i.e., this code has the "quick-look" property. (Clearly, the first K received sequences alone provide the "quick-look" for canonic systematic codes.)

Consider the following realizable function:

$$\frac{y(D)}{x(D)} = \frac{1 + D^{m+1}}{1 + D} = 1 + D + D^2 + \dots + D^m. \quad (145)$$

From (145) it follows that

$$y(D) + Dy(D) = x(D) + D^{m+1} x(D) \quad (146)$$

or

$$y(D) = x(D) + D \left[y(D) + D^m x(D) \right]. \quad (147)$$

A linear sequential circuit (LSC) which realizes equation (147) is shown in Figure 6.4. If $\bar{G}^{(j)}(D)$ is the complement of $G^{(j)}(D)$, then

$$y^{(j)}(D) = x(D) G^{(j)}(D) = x(D) \left[\bar{G}^{(j)}(D) + 1 + D + \dots + D^m \right]. \quad (148)$$

A circuit which realizes equation (148) is shown in Figure 6.5. Hence if $G^{(j)}(D)$ has a high density of 1's and therefore

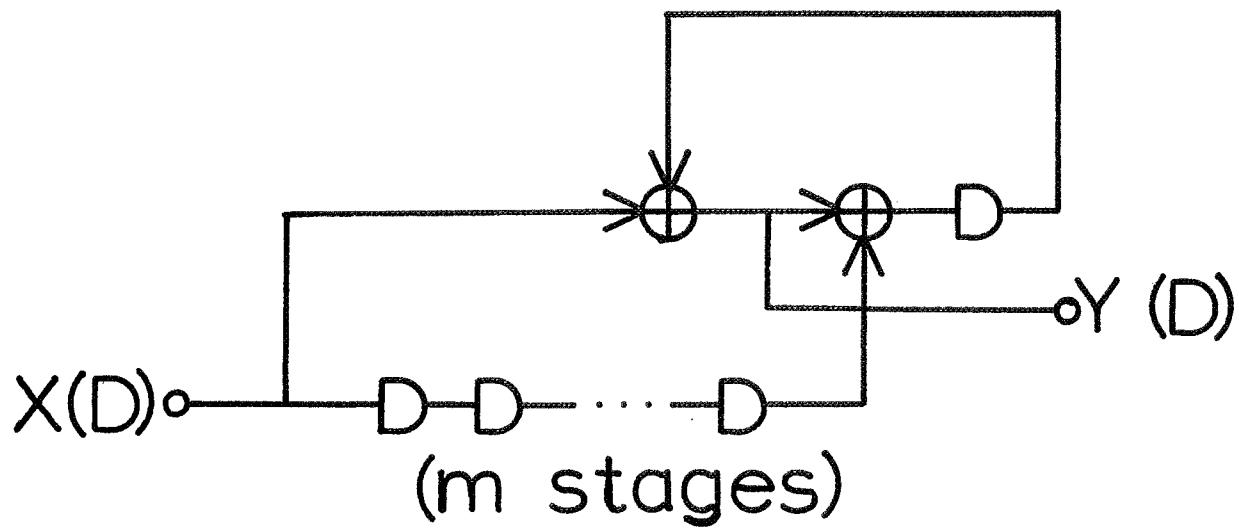


Fig. 6.4. An LSC which realizes equation (147).

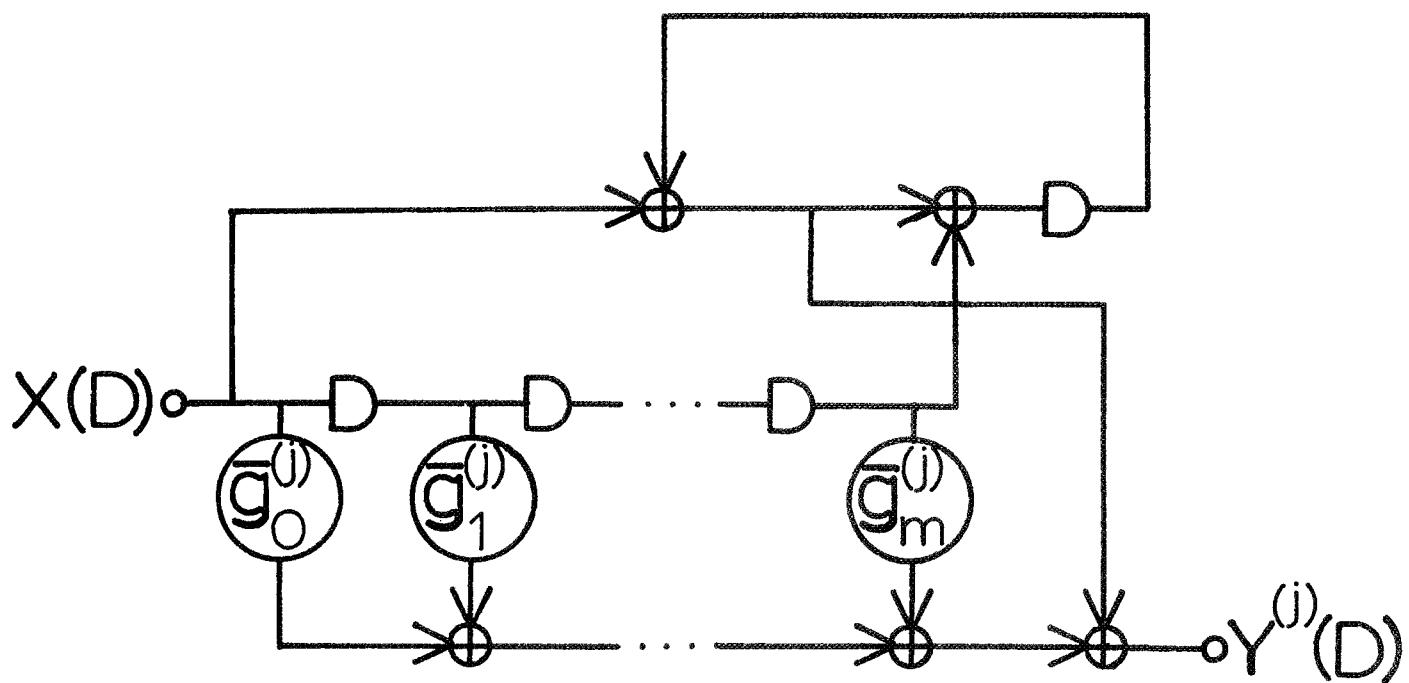


Fig. 6.5. An LSC which realizes equation (148)..

requires many modulo-2 adders in its realization, $\bar{G}^{(j)}(D)$, which requires few modulo-2 adders in its realization, can be implemented instead and the output complemented to produce $y^{(j)}(D)$. This fact will be used to greatly reduce the complexity of the class of $R = 1/2$ codes with $G^{(2)}(D) = D + G^{(1)}(D)$. A code of this type will now be constructed. (For convenience, let $g_{j1}^{(1)} = g_j^{(1)}$ and $g_{j1}^{(2)} = g_j^{(2)}$.)

Algorithm A9

- (0) Set $g_0^{(1)} = g_1^{(1)} = 1$, $d_1 = 3$, and $j = 2$.
- (1) Set $g_j^{(1)} \neq 0$.
- (2) Compute d_j . If $d_j > d_{j-1}$, go to (4).
- (3) Set $g_j^{(1)} = 1$.
- (4) If $j = m$, stop. Otherwise, set $j = j + 1$ and go to (1).

Note that the set of weights of the codewords $[y]$ with $x_0 \neq 0$ is the same whether $g_j^{(1)} = 0$ or $g_j^{(1)} = 1$ since complementing x_j cancels the effect of complementing $g_j^{(1)}$. Hence there is no need to recompute d_j in step (3) as in Algorithm A7 since if setting $g_j^{(1)} = 0$ does not increase d_j , then neither does setting $g_j^{(1)} = 1$. Also, an increase of two in d_j at any step is clearly impossible, since if $y_j = [1, 1]$ for some input sequence $[x]$, then $y_j = [0, 0]$ for the input sequence $[x']_j = [[x]_{j-1}, \bar{x}_j]$, and the column distance does not increase at all. Hence at each step in the algorithm, d_j either increases by 1 or stays the same. The code produced by Algorithm A9 with $m = 35$ is given in Table 6.11 along with the non-asymptotic Gilbert lower bound.

TABLE 6.11
CODES OBTAINED FROM ALGORITHM A9

j	$g_j^{(1)}$	$g_j^{(2)}$	d_j	d_G
0	1	1	2	2
1	1	0	3	3
2	1	1	3	3
3	0	0	4	4
4	1	1	4	4
5	1	1	4	4
6	0	0	5	4
7	1	1	5	5
8	1	1	5	5
9	1	1	5	5
10	0	0	6	5
11	1	1	6	6
12	0	0	7	6
13	1	1	7	6
14	1	1	7	6
15	0	0	8	7
16	1	1	8	7
17	1	1	8	7
18	1	1	8	7
19	1	1	8	8
20	0	0	9	8
21	1	1	9	8
22	1	1	9	8
23	1	1	9	9
24	1	1	9	9
25	1	1	9	9
26	0	0	10	9
27	1	1	10	9
28	1	1	10	10
29	1	1	10	10
30	0	0	11	10
31	0	0	11	10
32	1	1	11	11
33	1	1	11	11
34	1	1	11	11
35	1	1	11	11

$d_G = d_{\text{GILBERT BOUND}}$

The free distance for this code is known to be at least 17, and is probably much higher. The complete encoding circuit is shown in Figure 6.6. Note that only 11 modulo-2 adders are needed in the encoding circuit. This is exactly the same number needed to implement the $m = 35$ code of Algorithm A1 which has $d_{\text{FREE}} = 13$. Hence a substantial gain in d_{FREE} (and therefore in decoding probability of error) has been achieved without sacrificing anything in encoder complexity or "quick-look" capability. A truly surprising result!

Most of the codes presented in this section could have been easily extended out to about $m = 60$. However $m = 35$ seems to be a convenient length for many applications.

C. Performance of Codes with Sequential Decoding

1. Brief Description of the Simulated Sequential Decoder

In order to test the codes constructed in this chapter along with other known good codes, a sequential decoder was simulated on the Univac 1107 at the University Computer Center. Two simulations were made, one for a BSC and one for a Gaussian channel. Each program consists of four parts: a main program DECODE for reading in data and printing out results, a subprogram RANGEN for generating random noise, a subprogram TABSET for converting the random noise into tabular form suitable for the sequential decoder, and a subprogram SECO for the sequential decoding algorithm. Special thanks are due to Dr. K. Vairavan, who programmed both the RANGEN and TABSET subprograms, to Mr. John Geist and Mr. James Wruck

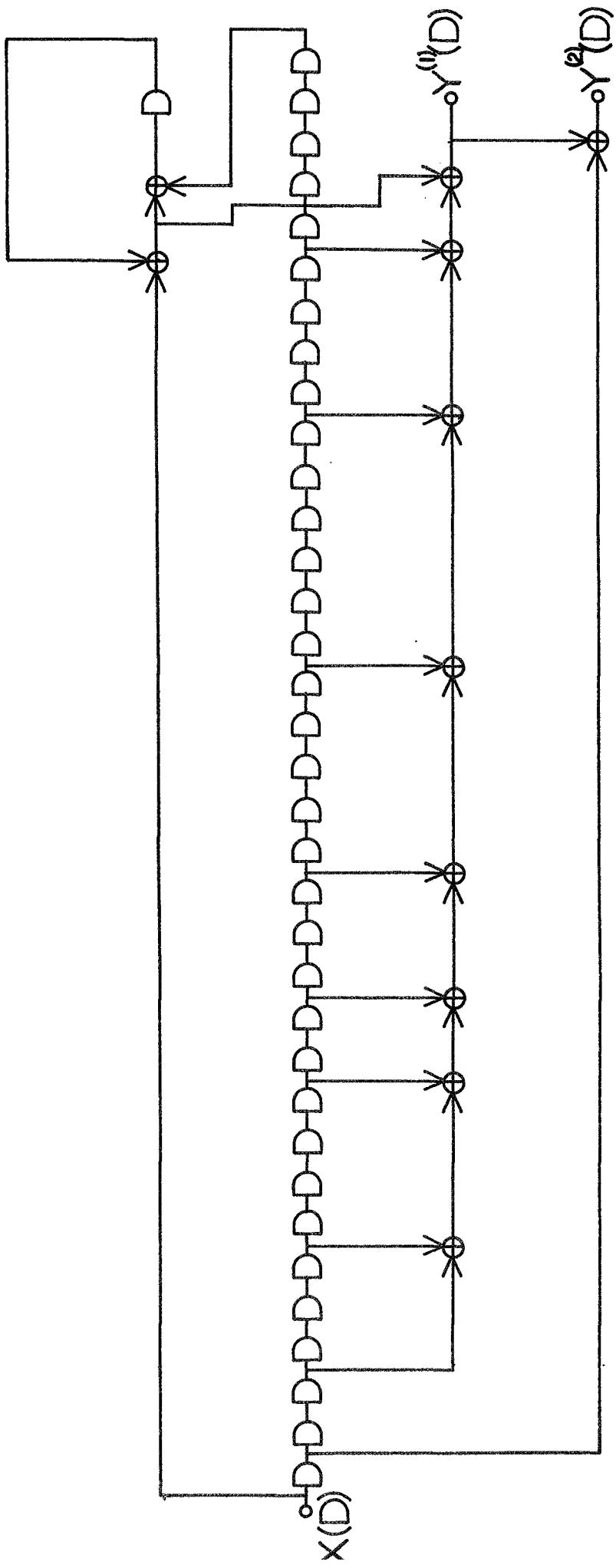


Fig. 6.6. The encoder for the code produced by algorithm A9.

for their numerous contributions to the efficiency of the programs, and to Mr. J. Chang and Mr. John Brennan for the preparation of the Gaussian program.

Each subprogram was written in assembly language to make the program as fast as possible, while the main program was written in FORTRAN to facilitate the input-output. Input information needed for the operation of the BSC program is as follows (for a complete discussion of sequential decoding parameters and notation, see Gallager [25]):

- (1) channel error probability p ;
- (2) the memory m of the code;
- (3) the generator of the code being tested;
- (4) the threshold increment H of the sequential decoding search;
- (5) a constant CONMET used to spread the difference between the metric values;
- (6) bins for the number of computations.

R_{comp} and the metric values are then computed from p and CONMET. The threshold increment H used in the production runs was determined experimentally. The value of H which optimizes the bound on computation is known to be 2 [3]. Since CONMET was chosen as 8, the "optimum" H is 16. However, through testing a single code for different values of H , it was determined that choosing H to be 32 was a better choice from both a computational and probability of error standpoint. These results are shown in Table 6.12.

TABLE 6.12

145

EFFECT OF VARYING THE THRESHOLD INCREMENT H

Code No. 1

Code Name Minimum Weight Code

Memory = 35

Rate = $\frac{1}{2}$

Type Systematic

Generator Sequences (Octal): 400000000000

651102104421

Known Distance Properties:

$d_{FD} = 13$

$d_{FREE} = 13$

Nature of Construction: Algorithm A1

Simulation Results:

(1) Channel BSC: $p = .033$ H 4 Total Frames 1000 Error Frames 10 Erased Frames 0

Computation:

N	292	310	350	400	475	550	700	1250	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	1000	1000	1000	999	986	607	154	42	15	4	0		

(2) Channel BSC: $p = .033$ H 8 Total Frames 1000 Error Frames 10 Erased Frames 0

Computation:

N	292	310	350	400	475	550	700	1250	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	1000	1000	991	957	756	226	55	16	5	0	0		

(3) Channel BSC: $p = .033$ H 16 Total Frames 1000 Error Frames 11 Erased Frames 0

Computation:

N	292	310	350	400	475	550	700	1250	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	1000	989	869	665	369	85	23	10	0	0	0		

(4) Channel BSC: $p = .033$ H 32 Total Frames 1000 Error Frames 16 Erased Frames 0

Computation:

N	292	310	350	400	475	550	700	1250	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	992	880	585	397	206	53	16	4	1	0	0		

(5) Channel BSC: $p = .033$ H 64 Total Frames 1000 Error Frames 26 Erased Frames 0

Computation:

N	292	310	350	400	475	550	700	1250	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	993	886	642	502	315	85	29	7	3	1	0		

(6) Channel BSC: $p = .033$ H 128 Total Frames 1000 Error Frames 177 Erased Frames 0

Computation:

N	292	310	350	400	475	550	700	1250	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	1000	997	982	954	891	565	227	86	22	9	1		

Each production run consisted of 1000 frames of 256 branches (blocks of information digits) each for a particular code and a particular channel error probability p . A frame was cut off and considered to be "erased" if it reached 50,000 computations. If a frame was decoded perfectly, it took $(256 + m)$ computations since 256 information blocks generate $(256 + m)$ transmitted blocks and the algorithm would count one computation for each correctly decoded block. Hence the computational bins are just numbers inclusive between $(256 + m)$ and 50,000 which record how many frames reached or exceeded that number of computations for decoding. Usually 13 computational bins were chosen for each production run.

In the Gaussian program the signal-to-noise ratio $\frac{E_b}{N_0}$ must be read in instead of p , where E_b is the energy per information digit and N_0 is the noise power spectral density. Then the procedure outlined in Jacobs [33] is followed to compute the metric values needed by the sequential decoder.

Output information available from the BSC program includes the following:

- (1) the actual branch metric values and R_{comp} ;
- (2) for each decoded frame:
 - (a) the number of computations;
 - (b) the number of decoding errors;
 - (c) the last branch decoded if the frame is erased;

- (d) the received sequence;
 - (e) the decoded sequence;
- (3) for the entire 1000 decoded frames:
- (a) the number of erased frames;
 - (b) the number of incorrectly decoded frames;
 - (c) the number of correctly decoded frames;
 - (d) the distribution of computation into bins.

Clearly the total number of error digits can be easily calculated from (2b). When the number of computations reached 50,000, decoding was terminated and the frame declared "erased". The output then recorded how far the search had progressed into the code tree when decoding was terminated. The printout of the received sequence and the decoded sequence for each frame is optional in the program.

For each computational bin, the number of frames which reached or exceeded that amount of computation is recorded. For example, the bin labeled 50,000 always contains the number of "erased" frames, and the bin labeled $(256 + m)$ always contains the total number of frames.

In the Gaussian program, additional output information about the channel is available.

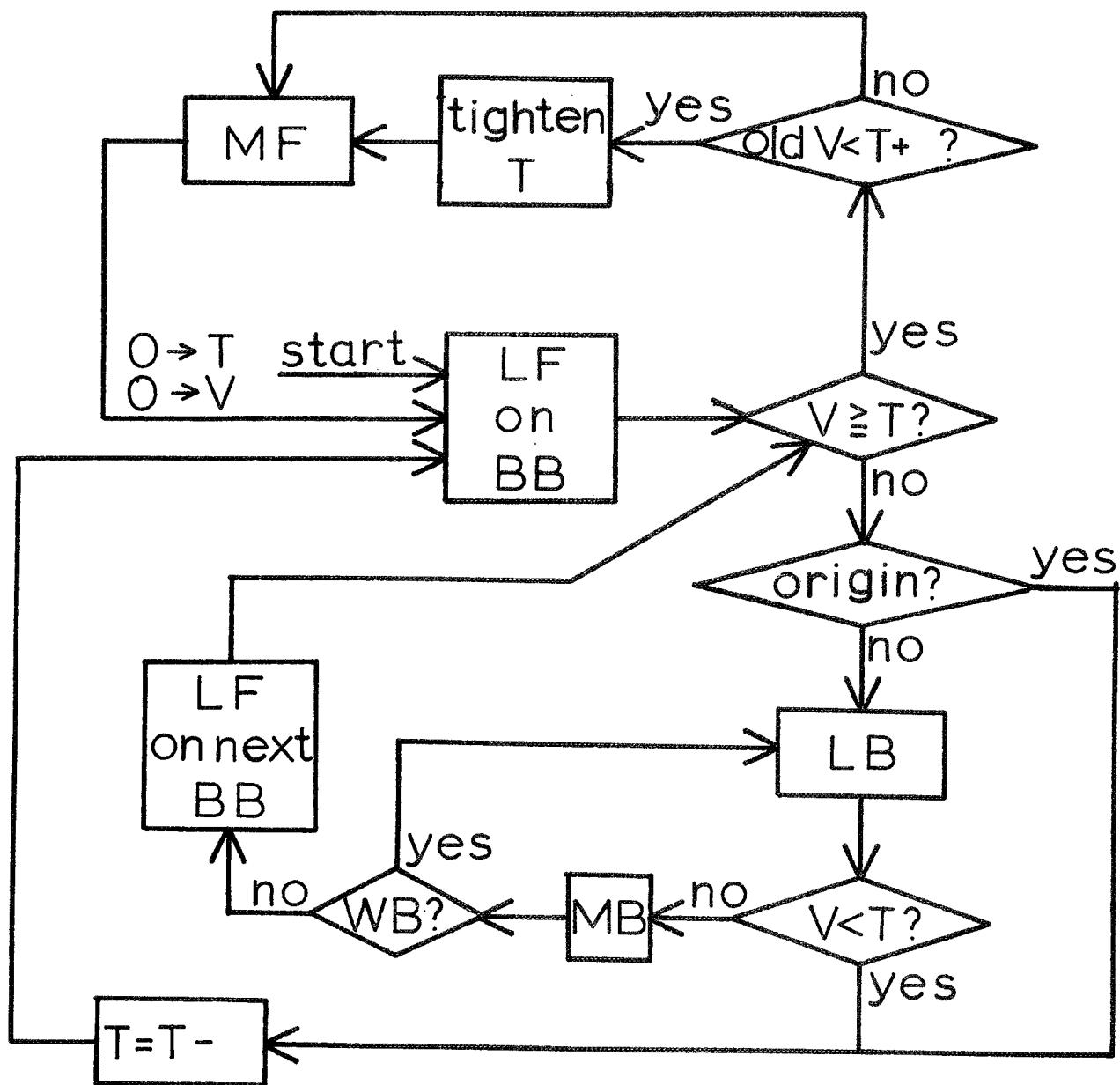
In the RANGEN subprogram, a library subroutine is used to generate a noise sequence distributed according to the channel error probability p for the BSC program. In the Gaussian program, the noise sequence is distributed according to the quantized channel model given by Jacobs [33].

TABSET merely converts the noise sequence into tabular data for use by SECO.

SECO is the actual sequential decoding algorithm. The version used is thoroughly discussed by Gallager [25]. A flow chart for SECO is shown in Figure 6.7. It is always assumed that the all-zero sequence has been transmitted. Since this was known to the programmer, SECO was always biased to look out on a 1 branch before looking out on a 0 branch in case the metric values on the two branches were tied. (Here the discussion pertains only to $R = \frac{1}{N}$ codes, in which there are only two branches emanating from each node in the code tree.) This undoubtedly resulted in slightly more computation than would be required normally, but of course this deficiency was common to all runs and would be expected to have no effect on the comparison between different codes.

A computation was counted as a "forward look", i.e., every time the decoder looked forward on a branch, and at no other time, a single computation was counted. Each computation, including the calculation of the parity digits, took about 100 μ sec of computer time.

The SECO algorithm is capable of handling both systematic and non-systematic codes with $m \leq 72$. Programs actually available are for $R = 1/2$, $R = 1/3$, and $R = 1/4$ only. However, only results on $R = 1/2$ codes will be reported here, since they are sufficiently representative of all rates.



LF = look forward BB = best branch V = node value

T = threshold H = threshold increment

MF = move forward LB = look back MB = move back

WB = worst branch

Fig. 6.7. SECO flow chart.

Also, data was taken for only three values of p and one value of $\frac{E_b}{N_0}$. These values are very typical, though, of a practical randomly distributed space channel. For $p = .033$, i.e., $R = 1/2 = (0.9) R_{\text{comp}}$, each production run of 1000 frames took about two minutes of computer time. For $p = .045$, i.e., $R = 1/2 = R_{\text{comp}}$, each run took about four minutes. For $p = .057$, i.e., $R = 1/2 = (1.1) R_{\text{comp}}$, each run took about 20 minutes. And for $\frac{E_b}{N_0} = 2$ or 3 db, each run took about five minutes.

2. Comparative Analysis of Codes

In Appendix A charts are given which have complete information on 13 different codes. A name and number is assigned to each code for identification purposes, and the means of construction for each code is briefly explained. Simulation results are given for the four channels described above. Not all the codes were tested with $p = .057$, since the computation time was so long.

An interesting comparison can be drawn between code 1 (from Algorithm A1) and code 3 (from Algorithm A6). Note that there are fewer error frames for code 3. This appears to be due to the fact that d_{FREE} is larger for code 3, since d_{FD} is the same for both codes, and substantiates the previous statement that d_{FREE} is a more important parameter than d_{FD} for sequential decoding.

Also compare code 11 (the non-systematic code from Algorithm A9) with code 12 (from Forney [28]). The non-systematic code is clearly superior in number of error

frames, although it has more erased frames. For the noisiest BSC, $p = .057$, code 11 makes no decoding errors while code 12 incorrectly decodes about 10 percent of the frames. However code 11 erases about 15 percent more frames than does code 12. But of these frames it appears that about half of them were incorrectly decoded by code 12. Massey [32] has termed this a "fools rush in where angels fear to tread" phenomenon. The slight computational advantage of code 12 over code 11 is clearly due to this phenomenon. Since code 11 is more easily implemented than code 12 and it has the "quick-look" property, the conclusion is that it is far superior to code 12 in system performance as well as system complexity. In fact, code 11 did not make a single decoding error in all four simulations. To the author's knowledge, code 12 is generally considered the best $m = 35$ systematic code available for sequential decoding. The performance of code 11 verifies the earlier statement that better results can be obtained for non-systematic codes than for systematic codes when used with sequential decoding (since more free distance is available for non-systematic codes).

Finally, compare the performance of code 2 with code 1. This indicates the advantage of using longer codes. However, encoder complexity increases with code length, which is an important consideration in many applications.

VII. Deriving Good Convolutional Codes

from Cyclic Codes

For completeness, this chapter will summarize attempts to derive good convolutional codes from good block codes, or at least to bound the distance of a convolutional code from known distance bounds on block codes. Some of these attempts have been quite successful, such as the codes due to McEliece [10] presented in Section VII.A, others relatively unsuccessful, such as those in Section VII.B, but still of some interest for their structure. In Section VII.D a possible new approach to this problem is discussed, and some limitations inherent in deriving convolutional codes from block codes are given in Section VII.E. In Section VII.C a method is presented for converting a code described by a parity-check matrix into a rationally equivalent generator matrix which is then reduced to Forney's [6] canonic generator matrix.

A. McEliece's Codes

The following result was obtained by McEliece [10] for fixed $R = 1/2$ systematic codes only. If $g(x)$ is the generator of a cyclic code with minimum distance d_g and $h(x)$ is the dual generator of $g(x)$ with minimum distance d_h , then the fixed $R = 1/2$ systematic code with $G^{(2)}(D) = g(D)$ has

$$d_{\text{FREE}} \geq \min \left[1 + d_g, 2 + d_h \right]. \quad (149)$$

Note that in order to assure a large free distance a cyclic code must be chosen which has a large minimum distance

and whose dual code also has large minimum distance. Hence very low rate cyclic codes are not a good choice, since their dual generators have very low distances.

Example 7.1 Let $g(x) = 1 + x^2 + x^5 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$, the dual generator of the Golay code. Then $d_g = 8$, $h(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$, and $d_h = 7$. Hence the fixed $R = 1/2$ systematic code with $G^{(2)}(D) = 1 + D^2 + D^5 + D^8 + D^9 + D^{10} + D^{11} + D^{12}$ has $d_{\text{FREE}} \geq 9$. Since the weight of the generator is 9, $d_{\text{FREE}} = 9$. |

Example 7.2 Let $g(x) = 1 + x^4 + x^5 + x^9 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{18} + x^{20} + x^{21} + x^{22} + x^{23}$, the generator of an $R = \frac{24}{47}$ cyclic code. $d_g = 11$, $h(x) = 1 + x^4 + x^5 + x^8 + x^9 + x^{11} + x^{12} + x^{15} + x^{18} + x^{20} + x^{23} + x^{24}$, and $d_h = 12$. Therefore the fixed $R = 1/2$ systematic code with $G^{(2)}(D) = g(D)$ has $d_{\text{FREE}} \geq 12$. But since all parity sequences p produced by an information sequence \underline{x} with $x_0 \neq 0$ either have weight at least 11 or \underline{x} has weight at least 12 and p has weight at least 2, the only possible weight 12 codeword is produced by the information sequence $\underline{x} = [1, 0, 0, \dots]$. But this codeword has weight 16. Hence $d_{\text{FREE}} \geq 13$. Also the only possible codewords of weight 13 are those with weight 2 information sequences. But it can easily be shown that no information sequences of weight 2 can produce a codeword of weight 13. Hence $d_{\text{FREE}} \geq 14$. But if the sequence whose transform is $h(D)$ is chosen as the information sequence, a codeword of weight 14 is obtained. Therefore $d_{\text{FREE}} = 14$. |

These codes are comparable to those constructed in Chapter VI, and it appears that long medium rate cyclic codes with good distance properties will produce good fixed $R = 1/2$ systematic convolutional codes for sequential decoding.

McEliece's result will now be extended to the important class of fixed $R = 1/2$ non-systematic codes and also to fixed $R = 1/3$ systematic codes. Let $g(x)$ and $h(x)$ be two relatively prime polynomials such that $g(x) h(x) = x^n - 1$. Clearly $g(x)$ and $h(x)$ each generate a cyclic code of length n , and they are each other's dual generators. Let d_g be the minimum distance of the cyclic code generated by $g(x)$ and let d_h be the minimum distance of the cyclic code generated by $h(x)$.

Now consider the fixed $R = 1/2$ non-systematic convolutional code with $G^{(1)}(D) = g(D)$ and $G^{(2)}(D) = h(D)$. Using the Euclidean division algorithm on an arbitrary information sequence \underline{x} whose transform is $x(D)$ implies that

$$x(D) = h(D) q_1(D) + r_1(D) = g(D) q_2(D) + r_2(D). \quad (150)$$

Therefore

$$y^{(1)}(D) = x(D) G^{(1)}(D) = q_1(D) (D^n - 1) + r_1(D) g(D) \quad (151)$$

$$y^{(2)}(D) = x(D) G^{(2)}(D) = q_2(D) (D^n - 1) + r_2(D) h(D)$$

and, since the remainder cannot have higher weight than the dividend after division by $D^n - 1$ for any n ,

$$w_H \left[y^{(1)}(D) \right] + w_H \left[y^{(2)}(D) \right] \geq w_H \left[r_1(D) g(D) \right] + \\ w_H \left[r_2(D) h(D) \right] , \quad (152)$$

where the Hamming weight of a polynomial is defined to be the number of non-zero terms in the polynomial. Then, since $\deg[r_1(D)] < \deg[h(D)]$ and $\deg[r_2(D)] < \deg[g(D)]$, $r_1(D) g(D)$ has weight at least d_g , $r_2(D) h(D)$ has weight at least d_h , and

$$d_{\text{FREE}} \geq d_g + d_h \quad (153)$$

unless $r_1(D) = 0$, $r_2(D) = 0$, or $r_1(D) = r_2(D) = 0$.

Assume $r_1(D) = 0$, $r_2(D) \neq 0$. Then $w_H \left[y^{(2)}(D) \right] \geq d_h$ and $w_H \left[y^{(1)}(D) \right] \geq 2$ since all multiples of $D^n - 1$ must have weight at least 2. Hence

$$d_{\text{FREE}} \geq 2 + d_h . \quad (154)$$

Assume $r_1(D) \neq 0$, $r_2(D) = 0$. Similarly

$$d_{\text{FREE}} \geq 2 + d_g . \quad (155)$$

Assume $r_1(D) = r_2(D) = 0$. Then $h(D) q_1(D) = g(D) q_2(D) = x(D)$. Since $h(D)$ and $g(D)$ are relatively prime,

$$q_1(D) = g(D) \frac{q_2(D)}{h(D)} = g(D) f(D) , \quad (156)$$

$$q_2(D) = h(D) f(D) , \quad (157)$$

and

$$y^{(1)}(D) = f(D) g(D) (D^n - 1) \\ y^{(2)}(D) = f(D) h(D) (D^n - 1) . \quad (158)$$

Again using the Euclidean algorithm, let

$$f(D) = g(D) Q_1(D) + R_1(D) = h(D) Q_2(D) + R_2(D). \quad (159)$$

Then

$$\begin{aligned} y^{(1)}(D) &= Q_2(D)(D^{2n}-1) + R_2(D)g(D)(D^n-1) \\ y^{(2)}(D) &= Q_1(D)(D^{2n}-1) + R_1(D)h(D)(D^n-1), \end{aligned} \quad (160)$$

and

$$\begin{aligned} w_H[y^{(1)}(D)] &\geq 2d_g \text{ since degree } [R_2(D)g(D)] < n \\ w_H[y^{(2)}(D)] &\geq 2d_h \text{ since degree } [R_1(D)h(D)] < n \end{aligned} \quad (161)$$

implies that

$$d_{\text{FREE}} \geq 2(d_h + d_g) \quad (162)$$

unless $R_1(D) = 0$, $R_2(D) = 0$, or $R_1(D) = R_2(D) = 0$.

Assume $R_1(D) = 0$, $R_2(D) \neq 0$. Then

$$d_{\text{FREE}} \geq 2d_g + 2. \quad (163)$$

If $R_1(D) \neq 0$, $R_2(D) = 0$, then

$$d_{\text{FREE}} \geq 2d_h + 2. \quad (164)$$

And if $R_1(D) = R_2(D) = 0$, then $f(D) = g(D)Q_1(D) = h(D)$

$Q_2(D)$. Since $g(D)$ and $h(D)$ are relatively prime,

$$Q_1(D) = h(D) \frac{Q_2(D)}{g(D)} = h(D)F(D), \quad (165)$$

$$Q_2(D) = g(D)F(D), \quad (166)$$

and

$$\begin{aligned} y^{(1)}(D) &= F(D)g(D)(D^{2n}-1) \\ y^{(2)}(D) &= F(D)h(D)(D^{2n}-1). \end{aligned} \quad (167)$$

Again applying the Euclidean algorithm, $F(D)$ can be written as

$$F(D) = h(D) q_1'(D) + r_1'(D) = g(D) q_2'(D) + r_2'(D), \quad (168)$$

and

$$\begin{aligned} y^{(1)}(D) &= q_1'(D) (D^n - 1)^3 + r_1'(D) g(D) (D^{2n} - 1) \\ y^{(2)}(D) &= q_2'(D) (D^n - 1)^3 + r_2'(D) h(D) (D^{2n} - 1). \end{aligned} \quad (169)$$

Since $(D^n - 1)^3 \neq D^{3n} - 1$, it is not immediately clear how to bound d_{FREE} in this case. However, as is shown in Appendix B, if

$$\begin{aligned} y^{(1)}(D) &= Q_1(D) (D^n - 1)^k + R_1(D) g(D) (D^n - 1)^{k-1} \\ y^{(2)}(D) &= Q_2(D) (D^n - 1)^k + R_2(D) h(D) (D^n - 1)^{k-1} \end{aligned} \quad (170)$$

for any positive integer k , then

$$w_H [y^{(1)}(D)] \geq L d_g \text{ and } w_H [y^{(2)}(D)] \geq L d_h, \quad (171)$$

where $L = w_H [(D^n - 1)^{k-1}]$. This proves the following result.

Theorem 7.1 For the class of fixed $R = 1/2$ non-systematic codes defined above, $d_{\text{FREE}} \geq 2 + \min [d_g, d_h]$.

Note that there is very little improvement over the $R = 1/2$ systematic codes.

Now consider the fixed $R = 1/3$ systematic convolutional code with $G^{(1)}(D) = 1$, $G^{(2)}(D) = g(D)$, and $G^{(3)}(D) = h(D)$. Following the same line of proof used to derive theorem 7.1 and again employing the results of Appendix B, it is easy to arrive at the following result.

Theorem 7.2 For the class of fixed $R = 1/3$ systematic convolutional codes defined above, $d_{\text{FREE}} \geq 2 + \min [d_g + d_h - 1, 2d_h, 2d_g]$.

Example 7.3 Again consider the Golay code and let

$$G^{(1)}(D) = 1, \quad G^{(2)}(D) = 1 + D^2 + D^4 + D^5 + D^6 + D^{10} + D^{11}, \text{ and}$$

$$G^{(3)}(D) = 1 + D^2 + D^5 + D^8 + D^9 + D^{10} + D^{11} + D^{12}. \quad \text{Then}$$

$$d_{\text{FREE}} \geq 16.$$

The extension of these results to other rates appears to be unrewardingly tedious. Furthermore, $R = 1/2$ and $R = 1/3$ codes are of the most practical interest. Equation (149) and theorem 7.2 guarantee that good long fixed $R = 1/2$ and $R = 1/3$ systematic codes can be found for use with sequential decoding. The result of theorem 7.1 for fixed $R = 1/2$ non-systematic codes, however, is somewhat disappointing.

B. Wyner-Sullivan Codes

A class of very high rate fixed codes with $d_{FD} = 5$, based on the structure of BCH codes, was discovered by Wyner [34]. Sullivan [35] has introduced simple encoding and algebraic decoding procedures for these codes and has shown that they do not exhibit the error propagation effect [36] characteristic of certain convolutional codes.

Consider a parity check matrix of the following form:

$$\underline{H} = \begin{bmatrix} \alpha^{N-1} & \alpha^{N-2} & \dots & \alpha & 1 & 0 & 0 & \dots & 0 & 0 \\ (\alpha^3)^{N-1} & (\alpha^3)^{N-2} & \dots & \alpha^3 & 1 & \alpha^{N-1} & \alpha^{N-2} & \dots & \alpha & 1 \dots \\ 0 & 0 & \dots & 0 & 0 & (\alpha^3)^{N-1} & (\alpha^3)^{N-2} & \dots & \alpha^3 & 1 \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & 0 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & & \cdot \\ & & & & & \cdot & \cdot & & \cdot & \cdot \\ & & & & & \cdot & \cdot & & \cdot & \cdot \end{bmatrix}, \quad (172)$$

where α is a primitive element of $GF(2^r)$. Since each power of α can be expressed as a binary r -tuple, each successive block of parity check digits is shifted down r rows. Hence there are r parity checks per time unit and $2^r - 1$ distinct non-zero powers of α . Therefore $N-K = r$ and $R = \frac{2^r - r - 1}{2^r - 1}$.

The convolutional code defined by an \underline{H} matrix is taken to be precisely the set of sequences in the null space of \underline{H}^T , and no other sequences. Wyner showed that $d_{FD} = 5$ for all the convolutional codes in this class by noting that the first $2r$ rows of (172) are also a parity check matrix for a double error correcting BCH code. (Wyner and Ash [37] have defined d_{FD} as the minimum number of columns of \underline{H} which can add to zero, including at least one column from the first block.)

The decoder looks at three blocks of syndrome digits before estimating a block of error digits. Hence the decoding memory $\bar{m} = 2$ and the decoding constraint length $\bar{n}_A = 3N = 3(2^r - 1)$. Clearly as $r \rightarrow \infty$, $R \rightarrow 1$ and in the limit infinite constraint length $R = 1$ codes with $d_{FD} = 5$ are approached. Sullivan [35] has shown that d_{FD} for all these codes lies well above the non-asymptotic Gilbert lower bound on d_{FD} .

The chief disadvantage of the Wyner-Sullivan codes is that no method has yet been found to extend the results to other than $\bar{m} = 2$, $d_{FD} = 5$ codes. Clearly this is a serious limitation on the usefulness of the codes.

C. Forney Canonic Form

Forney [6] has outlined an interesting procedure for converting an encoder into what he calls a minimal canonic encoder, which is rationally equivalent to the original encoder. An example will be given for the Wyner-Sullivan code with $r = 3$ which is described by the following parity check matrix:

$$\underline{H}_m = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (173)$$

This is an $R = \frac{4}{7}$ code with $d_{FD} = 5$, $\bar{m} = 2$, and $\bar{n}_A = 21$.

The first step is to write the parity-check matrix \underline{H} as an $(N-K) \times N$ matrix of parity-check functions $\underline{H}(D)$. This results in the following matrix:

$$\underline{H}^T(D) = \begin{bmatrix} 1+D & D & 1 \\ 1 & 1+D & 1 \\ 1+D & 1+D & D \\ D & 1 & 1 \\ 1+D & 0 & D \\ 0 & 1+D & D \\ 0 & 0 & 1+D \end{bmatrix} \quad (174)$$

Then convert $\underline{H}(D)$ into canonic systematic form by performing elementary column operations on $\underline{H}^T(D)$ which do not change its null space. This results in the following matrix:

$$\underline{H}^T(D) = \begin{bmatrix} 1 & D/1+D & 1/1+D^2 \\ 1/1+D & 1 & 1+D+D^2/1+D \\ 1 & 1 & D/1+D \\ D/1+D & 1/1+D & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (175)$$

Now $\underline{H}(D)$ can be converted to the generator matrix $\underline{G}(D)$, as detailed in Wyner [38], where

$$\underline{G}(D) = \begin{bmatrix} 1 0 0 0 1 & D/1+D & 1/1+D^2 \\ 0 1 0 0 1/1+D & 1 & 1+D+D^2/1+D^2 \\ 0 0 1 0 1 & 1 & D/1+D \\ 0 0 0 1 D/1+D & 1/1+D & 1 \end{bmatrix} \quad (176)$$

Multiplying each row of $\underline{G}(D)$ by the least common multiple of its denominator terms results in the following matrix:

$$\underline{G}(D) = \begin{bmatrix} 1+D^2 & 0 & 0 & 0 & 1+D^2 & D+D^2 & 1 \\ 0 & 1+D^2 & 0 & 0 & 1+D & 1+D^2 & 1+D+D^2 \\ 0 & 0 & 1+D & 0 & 1+D & 1+D & D \\ 0 & 0 & 0 & 1+D & D & 1 & 1+D \end{bmatrix} \quad (177)$$

In Forney's [6] procedure, the greatest common denominator of the determinants of all the $K \times K$ submatrices of $\underline{G}(D)$

is calculated. This is found to be $(1+D)^2$. Then a rationally equivalent matrix which is a basis for the row space of the above matrix can be constructed. This leads to the following matrix:

$$\underline{G}(D) = \begin{bmatrix} 1+D & 1+D & 0 & 0 & D & 1 & D \\ 0 & 1+D & 1 & 0 & 0 & D & 1+D \\ 0 & 0 & 1+D & 0 & 1+D & 1+D & 0 \\ 0 & 0 & 0 & 1+D & D & 1 & 1+D \end{bmatrix} \quad (178)$$

Finally $\underline{G}(D)$ is converted to what Forney [6] calls the minimal canonic matrix which is rationally equivalent to each of the above matrices. This results in the following matrix:

$$\underline{G}(D) = \begin{bmatrix} 1+D & 0 & 1 & 0 & D & 1+D & 1 \\ 0 & 1+D & 1 & 0 & 0 & D & 1+D \\ 0 & 0 & 1+D & 0 & 1+D & 1+D & D \\ 0 & 0 & 0 & 1+D & D & 1 & 1+D \end{bmatrix} \quad (179)$$

Note that Forney's canonic form does not necessarily mean the code is in systematic form.

Hence the Wyner-Sullivan code with $r = 3$ has a rationally equivalent encoder with encoding memory $m = 1$ and encoding constraint length $n_A = 14$. Therefore the encoding circuit for the code can be implemented with only one-stage shift registers.

D. Codes Dependent on the Length of Information Sequence Bound

In Section VII.A, it was shown that the free distance of certain convolutional codes depends on the cyclic code properties of the polynomials in the matrix $\underline{G}(D)$. In this section a different view of this relationship will be taken. Consider an $R = 1/2$ systematic fixed code with $\underline{G} = \begin{bmatrix} I_\infty & \underline{Q} \end{bmatrix}$, where I_∞ is a semi-infinite identity matrix and

$$\underline{Q} = \begin{bmatrix} g_0^{(2)} & g_1^{(2)} & g_2^{(2)} & \dots & g_m^{(2)} \\ g_0^{(2)} & g_1^{(2)} & \dots & g_{m-1}^{(2)} & g_m^{(2)} \\ g_0^{(2)} & \dots & g_{m-2}^{(2)} & g_{m-1}^{(2)} & g_m^{(2)} \\ g_0^{(2)} & & g_1^{(2)} & g_2^{(2)} & \dots & g_m^{(2)} \end{bmatrix}. \quad (180)$$

Note that (180) truncated after k rows could also serve as the generator matrix of an (n, k) cyclic code. Therefore if the length of the information sequence needed to produce the shortest minimum free weight codeword is known to be less than or equal to k , d_{FD} of the cyclic code provides a lower bound on d_{FREE} of the convolutional code. Hence if the length of information sequence bound were known to be m for $R = 1/2$ fixed systematic codes (cf. Chapter V), d_{FD} for all cyclic codes with generator $g(x)$ and $R > 1/2$ would provide a lower bound on d_{FREE} for the fixed $R = 1/2$ systematic convolutional code with $G^{(2)}(D) = g(D)$. This would

mean that the BCH bound on minimum distance [39] could be applied to convolutional codes as well as to block codes. But, of course, this conjecture has yet to be proved.

E. Some Upper Bounds on Free Distance

Besides providing lower bounds on the distance properties of certain convolutional codes, cyclic codes also provide upper bounds on distance for certain convolutional codes.

Theorem 7.3 Given an $R = 1/2$ systematic fixed code with $G^{(2)}(D) = G(D) = 1 + g_1 D + \dots + D^m$ and $w_H[G(D)] > 2$, then

$$(1) \quad d_{FD} \leq 1 + w_H[H(D)],$$

$$(2) \quad d_{FREE} \leq 2 + w_H[H(D)],$$

where $H(D)$ is the lowest degree polynomial such that $H(D) \mid G(D) = D^n - 1$.

Proof

- (1) Let \underline{h} be the information sequence whose transform is $H(D)$. Clearly $h_0 = 1$ and $w_H[\underline{h} \underline{G}] = 1 + w_H(\underline{h})$. Therefore $d_{FD} = d_m \leq 1 + w_H[H(D)]$.
- (2) $w_H(\underline{h} \underline{G}) = 2 + w_H(\underline{h})$. Therefore $d_{FREE} \leq 2 + w_H[H(D)]$. |

Theorem 7.3 indicates that generators of low rate cyclic codes make poor choices of $G^{(2)}(D)$ for $R = 1/2$ systematic fixed codes. However, since the maximal length polynomials are "pseudo random", it might be inferred that they would

make very good choices for $G^{(2)}(D)$ because Shannon [31] showed that a randomly constructed code has a very high probability of being a good code. Example 7.4 will prove this conjecture to be incorrect. However the distance properties of the other "pseudo random" generator sequences are still open to investigation.

Example 7.4 Consider the class of $R = 1/2$ fixed systematic codes with $G^{(2)}(D) = g(D)$, where $g(x)$ is the generator of a maximal length code, given in Table 7.1.

TABLE 7.1

<u>Rate of the Maximal Length Code</u>	<u>Memory of the Convolutional Code</u>	<u>Upper Bound on the Free Distance</u>
2/3	1	5
3/7	4	6
4/15	11	7
5/31	26	8
6/63	57	9
7/127	120	10

Clearly, the convolutional codes get very bad as the rate of the maximal length codes decreases.]

Now a theorem will be given whose proof is similar to the proof of theorem 7.3(2), and hence will be omitted.

Theorem 7.4 Given an $R = 1/2$ systematic fixed code with $G^{(2)}(D) = G(D) + L(D)$, where $G(D)$ is a polynomial and

$L(D)$ is a very low weight polynomial, then $d_{\text{FREE}} \leq 2 + \left\{ w_H [L(D)] + 1 \right\} w_H [H(D)]$, where $H(D)$ is the lowest degree polynomial such that $G(D) H(D) = D^n - 1$. |

Theorem 7.4 provides a tight bound on d_{FREE} for those codes whose generator is very close to the generator of a code tightly bounded by theorem 7.3.

The results of theorem 7.3 and theorem 7.4 can be extended to fixed systematic codes of all rates and to fixed non-systematic codes, but the bounds are the tightest in the $R = 1/2$ systematic case.

VIII. Summary, Conclusions, and Recommendations for Further Research

In Chapter I, the formalism for convolutional encoding was introduced and various problems associated with convolutional codes were discussed. Two different notational systems were presented, the D-transform approach introduced by Massey [4] and extended by Forney [6], which is used only for fixed codes, and an extension of the vector notation of Wozencraft and Reiffen [3], which is used for both fixed and time-varying codes.

From Chapter II onward, only binary codes, i.e., codes defined over GF(2), were considered, primarily for convenience since many of the results are easily extended to non-binary codes. In Chapter II, feedback decoding minimum distance, definite decoding minimum distance, free distance, and reverse distance were defined for both fixed and periodic codes. Also two new distance measures, column distance and row distance were introduced. It was claimed that d_{FREE} is a more appropriate distance measure for sequential decoding than d_{FD} , a fact verified in later chapters. A general definition of definite decoding minimum distance, valid for systematic and non-systematic codes of all rates, is still being sought. In Chapter IV, it was necessary to give an alternate definition of n_{DD} and d_{DD} in order to obtain a lower bound on d_{DD} for non-systematic periodic codes.

In Chapter III, a new definition of encoder equivalence, called causal equivalence, was given which not only guarantees that the encoders produce the same output sequences, but that they have the same column distance properties when only causal information sequences are allowed. This is slightly different from Forney's [6] definition, termed rational equivalence here, where two equivalent encoders can have different values of column distance. Then a method was given whereby a non-systematic encoder with no causally equivalent systematic encoder can be simply converted to a systematic encoder without decreasing column distance. Such an encoder is called causally dominant to the original encoder. Hence systematic encoders are always as good as non-systematic encoders as far as distance properties are concerned. However, if a non-systematic polynomial encoder is converted to a causally dominant or a causally equivalent systematic encoder with rational functions and then each generator function is truncated to preserve the encoding memory, d_{FD} is preserved but d_{FREE} is not. Finally, general parity-check matrices and syndrome forming circuits were obtained for non-systematic encoders of all rates.

In Chapter IV, bounds were obtained on various distance measures. A lower bound on d_{DD} was shown for both systematic and non-systematic periodic codes. It is conjectured that these bounds hold also for fixed codes, but this remains unproved. Also a Gilbert lower bound on d_{FD} was given for a

simply implemented subclass of periodic codes with period $T = 2m + 1$. A strong lower bound on d_{FREE} was obtained for non-systematic periodic codes, and the result was used to prove an upper bound on error probability for maximum likelihood decoding over a BSC which is superior to Viterbi's [2] upper bound for low rates and which meets Viterbi's lower bound at $R = 0$. An upper bound on d_{FREE} for all fixed codes which is essentially the same as McEliece and Rumsey's [10] bound for fixed $R = \frac{1}{N}$ systematic codes was also shown. It was indicated how this result can also be extended to periodic codes. Finally, a Gilbert lower bound on a simply implemented subclass of fixed $R = 1/2$ non-systematic codes was obtained. However, a complete set of bounds on the distance properties of convolutional codes, most noticeably a Hamming upper bound on d_{FD} , is still missing.

A major consequence of the results of Chapter IV was the demonstration that more free distance is available from non-systematic codes than from systematic codes of the same length. Hence it was conjectured that non-systematic codes would perform better with sequential decoding than systematic codes of the same constraint length, a fact verified experimentally as reported in Chapter VI.

In Chapter V, some partial results were presented on the still outstanding problem of bounding the length of the information sequence needed to produce the minimum free weight codeword. Also some methods of calculating the free

distance were given, a problem which would be greatly simplified by obtaining a tight length of information sequence bound.

In Chapter VI, simple and efficient algorithms were given for constructing fixed systematic convolutional codes with d_{FD} considerably larger than the non-asymptotic Gilbert lower bound out to $m = 71$ for $R = 1/2$, and $m = 35$ for $R = 1/3$ and $R = 1/4$. The algorithms always retained the property of minimizing the number of modulo-2 adders needed in the obvious encoding circuit for codes of a given length and minimum distance, an important consideration in many applications. In addition, an algorithm for constructing fixed systematic $R = 1/2$ codes with known d_{FREE} was presented.

Other algorithms for constructing $R = 1/2$ fixed codes which do not minimize the number of modulo-2 adders in the encoding circuit were also given. In particular a construction algorithm was presented for the class of fixed $R = 1/2$ non-systematic codes for which a Gilbert lower bound was obtained in Chapter IV. These codes have been shown by Massey [32] to possess the "quick-look" property of systematic codes, and to have a very simple encoding circuit. The code of this type which was constructed was shown to have a very large d_{FREE} , a consequence of its being a non-systematic code. A very strong result would be obtained if any of the algorithms presented in Chapter VI could be shown to produce good codes of arbitrary length.

Finally, a brief description of the simulated sequential decoding program was given, and a comparative analysis of sequential decoding performance for various codes with memory 35 was presented. Four channels, a Gaussian channel with $\frac{E_b}{N_0} = 2.0$ and three BSC's with $R = 1/2 = 0.9 R_{\text{comp}}$, $R = 1/2 = R_{\text{comp}}$, and $R = 1/2 = 1.1 R_{\text{comp}}$, respectively, were simulated. The fixed $R = 1/2$ non-systematic code was far superior to all the other codes tested as regards system performance. Indeed it did not make a single decoding error over any of the four simulated channels. This was due, of course, to the advantage in free distance which non-systematic codes possess over systematic codes. Also, because of its simple implementation, it was very desirable as far as system complexity is concerned.

In Chapter VII, McEliece's [10] result obtaining fixed $R = 1/2$ systematic codes with large d_{FREE} from cyclic codes was extended to fixed $R = 1/3$ systematic codes and fixed $R = 1/2$ non-systematic codes. The results for the latter case were disappointing, but good fixed $R = 1/3$ systematic codes were found. Some codes discovered by Wyner [34] and Sullivan [35] were presented via the parity-check matrix, and a method was given for converting a parity-check matrix in non-systematic form into a rationally equivalent generator matrix. However the codes considered have not been generalized beyond $\bar{m} = 2$ and $d_{\text{FD}} = 5$. A new approach to constructing good convolutional codes from cyclic codes was introduced, but it was seen to depend on proving a tight

length of information sequence bound. Finally, some limitations on constructing convolutional codes from block codes were presented. The problem of finding some construction technique which yields good codes as the memory gets arbitrarily large remains unsolved for convolutional codes, as indeed it is still unsolved for block codes.

APPENDIX A

Code No. 2

Code Name Minimum Weight Code

Memory = 71

Rate = $\frac{1}{2}$ Type Systematic

Type Systematic

Generator Sequences (Octal): 4000000000000000000000000000

651102104421022041101101

Known Distance Properties:

$$d_{FD} = 21$$

$$d_{\text{FREE}} = 21$$

Nature of Construction: Algorithm A1

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 9 Erased Frames 4
 (1) Channel Gauss:

Computation:										Total Error Bits: 26				
N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K	
# Frames with #C = N	1000	995	970	904	727	607	455	372	282	212	57	13	5	

(2) Channel BSC: p = .033 H 32 Total Frames 1000 Error Frames 11 Erased Frames 0

Computation:										Total Error Bits: 16				
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with $\#C \geq N$	1000	978	507	249	144	81	37	23	16	5	1	0	0	

(3) Channel BSC: p = .045 H 32 Total Frames 1000 Error Frames 62 Erased Frames 1

Computation:													Total Error, Bits:320			
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	.	.	
# Frames with $\#C \geq N$	1000	1000	832	589	458	363	217	146	103	29	11	5	1			

(4) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(5) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(6) Channel	H	Frames	Frames	Frames
Computation:	Total Error Bits:			
N				
# Frames				

Code No. 5

Code Name Maximum Weight Code

Memory = 35

Rate = $\frac{1}{2}$ Type Systematic

Generator Sequences (Octal): 400000000000

736677773575

Known Distance Properties:

$$d_{FD} = 13$$

$$d_{\text{FREE}} = 16$$

Nature of Construction: Algorithm A7

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 3 Erased Frames 2
(1) Channel Gauss: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 3 Erased Frames 2

Computation: Total Error Bits: 10

N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K		
# Frames with #C ≥ N	1000	974	902	809	652	539	414	344	278	202	65	26	8		

(2) Channel BSC: p = .033 H 32 Total Frames 1000 Error Frames 0 Erased Frames 0

Computation: Total Error Bits: 0

N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with #C \geq N	1000	878	372	195	121	89	36	19	17	9	3	0	0	

(3) Channel BSC: p = .045 H 32 Total Frames 1000 Error Frames 8 Erased Frames 5

Computation: Total Error_Bits: 40

N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with #C ≥ N	1000	992	763	534	415	327	205	140	110	56	25	15	5		

Computation: Total Error Bits:

N
Frames
with $\#C \geq N$

(5) Channel H Total Frames Error Frames Erased Frames

Computation: Total Error Bits:

(6) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

Computation: Total Error Bits:

Code No. 7

Code Name Balanced Code

Memory = 35

Rate = $\frac{1}{3}$ Type Systematic

Generator Sequences (Octal): 400000000000

653125446515

Known Distance Properties:

$$d_{\text{FD}} = 13$$

16 ~~is~~ d_{FREE} ~~is~~ 20

Nature of Construction: Algorithm A8

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 3 Erased Frames 5
 (1) Channel Gauss: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 3 Erased Frames 5

Computation: Total Error Bits: 33

N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K		
# Frames with #C = N	1000	971	908	814	665	537	410	399	263	193	68	18	10		

(2) Channel BSC: p = .033 H 32 Total Frames 1000 Error Frames 0 Erased Frames 0

Computation: Total Error Bits: 0

N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with #C = N	1000	880	378	193	126	85	33	21	15	10	3	0	0		

(3) Channel BSC: p = .045 H 32 Total Frames 1000 Error Frames 12 Erased Frames 2

Computation: Total Error.Bits: 103

N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with #C = N	1000	991	743	516	417	344	206	151	116	58	25	14	2		

(4) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

Computation: _____ **Total Error Bits:** _____

(5) Channel H Total Frames _____ Error Frames _____ Erased Frames _____

Computation: Total Error Bits:

(6) Channel H Total Frames Error Frames Erased Frames

Computation: Total Error Bits:

Code No. 8

Code Name Balanced Adjoint

Memory = 35

Rate = $\frac{1}{3}$ Type Systematic

Generator Sequences (Octal): 400000000000

732453703632

Known Distance Properties:

$$d_{FD} = 13$$

$$18 \leq d_{\text{FREE}} \leq 22$$

Nature of Construction: Adjoint of code no. 7

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 3
 (1) Channel Gauss:

Computation:													Total	Error	Bits:	0
N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K			
# Frames with $\#C \geq N$	1000	973	900	800	649	536	412	324	266	211	59	21	8			

(2) Channel BSC: $p = .033$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 0

Computation:											Total Error Bits: 0			
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with $\#C \geq N$	1000	873	376	191	114	87	37	20	15	7	2	0	0	

(3) Channel BSC: $p = .045$ H 32 Total Frames 1000 Error Frames 5 Erased Frames 6

Computation:													Total Error Bits: 64		
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with #C \leq N	1000	992	746	523	410	334	203	144	116	54	30	15	6		

(4) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(5) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(6) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

Code No. 9

Code Name Balanced Bussgang 2 Code

Memory = 35 Rate = $\frac{1}{2}$ Type Systematic

Rate = $\frac{1}{2}$ Type Systematic

Generator Sequences (Octal): 400000000000

732443151623

Known Distance Properties:

$$d_{\text{FD}} = 13$$

$$18 \leq d_{\text{FREE}} \leq 20$$

Nature of Construction: The code obtained by using Algorithm A8 to extend one of Bussgang's optimal codes.

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 1 Erased Frames 3
(1) Channel Gauss: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 1 Erased Frames 3

Computation:												Total Error Bits: 7		
N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K	
# Frames with #C = N	1000	973	899	799	648	521	395	326	262	199	57	19	10	

(2) Channel BSC: $p = .033$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 0

Computation:													Total Error Bits: 0		
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with #C \leq N	1000	873	371	181	114	78	30	17	14	7	2	0	0		

(3) Channel BSC: p = .045 H 32 Total Frames 1000 Error Frames 4 Erased Frames 3

Computation:													Total Error	Bits: 38	
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	992	739	512	391	307	192	133	105	43	25	13	3		

(4) Channel _____ . H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(5) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(6) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

Code No. 11

Code Name Non-systematic Code

Memory = 35

Rate = $\frac{1}{3}$ Type Non-systematic

Generator Sequences (Octal): 733533676737

533533676737

Known Distance Properties:

$$d_{FD} = 11$$

$$17 \leq d_{\text{FREE}}$$

Nature of Construction: Algorithm A9

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 5
 (1) Channel Gauss: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 5

Computation:										Total Error Bits: 0				
N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K	
# Frames with #C = N	1000	968	909	835	676	567	445	358	292	225	70	17	9	

(2) Channel BSC: $p = .033$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 0

Computation:										Total Error Bits: 0				
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with #C ≥ N	1000	883	405	223	135	92	47	26	18	5	2	0	0	

(3) Channel BSC: $p = .045$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 8

Computation:										Total Error Bits: 0				
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with #C \geq N	1000	991	785	581	477	382	240	167	134	63	36	23	8	

(4) Channel BSC: $p = .057$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 249

Computation:												Total Error Bits: 0		
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with $\#C \geq N$	1000	1000	949	863	802	753	640	585	543	440	358	303	249	

(5) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

(6) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

Code No. 12

Code Name NASA Code

Memory = 35

$$\text{Rate} = \frac{1}{2}$$

Type Systematic

Generator Sequences (Octal): 400000000000

715473701317

Known Distance Properties:

$$d_{FD} = 14$$

$$d_{\text{FREE}} = 18$$

Nature of Construction: The adjoint of the code Forney obtained by using the Lin-Lyne algorithm to extend one of Bussgang's optimal codes.

Simulation Results: $E_b/N_o = 2.0$ H 32 Total Frames 1000 Error Frames 0 Erased Frames 4

Computation:												Total Error Bits: 0		
N	292	400	450	500	600	700	850	1000	1200	1500	4000	10K	25K	
# Frames with #C >= N	1000	969	900	810	652	523	404	327	254	188	60	19	9	

(2) Channel BSC: p = .033 H 32 Total Frames 1000 Error Frames 0 Erased Frames 0

Computation:										Total Error Bits: 0				
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	
# Frames with $\#C \leq N$	1000	884	387	189	111	79	26	18	16	8	2	2	0	

(3) Channel BSC: p = .045 H 32 Total Frames 1000 Error Frames 2 Erased Frames 4

Computation:												Total Error Bits: 12			
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K	.	
# Frames with #C > N	1000	991	756	510	403	320	187	138	104	48	31	11	4		

(4) Channel BSC: $P = .057$ H 32 Total Frames 1000 Error Frames 87 Erased Frames 108

Computation:										Total Error Bits: 2071					
N	292	400	550	700	850	1000	1500	2000	2500	5000	10K	20K	50K		
# Frames with $\#C \geq N$	1000	1000	932	817	734	673	532	455	412	319	237	181	108		

(5) Channel	H	Total Frames	Error Frames	Erased Frames
-------------	---	--------------	--------------	---------------

(6) Channel _____ H _____ Total Frames _____ Error Frames _____ Erased Frames _____

APPENDIX B

Generalizing from equations (151), (160), and (169), $y^{(1)}(D)$ and $y^{(2)}(D)$ can always be written in the following form:

$$y^{(1)}(D) = Q_1(D) (D^n - 1)^k + R_1(D) g(D) (D^n - 1)^{k-1} \quad (B1)$$

$$y^{(2)}(D) = Q_2(D) (D^n - 1)^k + R_2(D) h(D) (D^n - 1)^{k-1}$$

for some positive integer k .

Clearly if k is a power of 2, then

$$(D^n - 1)^k = D^{kn} - 1 \quad (B2)$$

and

$$(D^n - 1)^{k-1} = D^{(k-1)n} + D^{(k-2)n} + D^{(k-3)n} + \dots + 1. \quad (B3)$$

Therefore

$$\begin{aligned} w_H [y^{(1)}(D)] &\geq k d_g \\ w_H [y^{(2)}(D)] &\geq k d_h \end{aligned} \quad (B4)$$

since

$$\begin{aligned} \text{degree } [R_1(D)] &< \text{degree } [h(D)] \\ \text{degree } [R_2(D)] &< \text{degree } [g(D)]. \end{aligned} \quad (B5)$$

Let $Q_1(D) = P_0(D) + P_1(D) D^n + P_2(D) D^{2n} + P_3(D) D^{3n} + \dots$, where $\text{degree } [P_i(D)] < n$ for all i . Then for $k = 3$,

$$y^{(1)}(D) = [P_0(D) + P_1(D) D^n + P_2(D) D^{2n} + \dots] [D^{3n} + D^{2n} + D^n + 1] + R_1(D) g(D) [D^{2n} + 1] \quad (B6)$$

$$\begin{aligned}
&= \left[P_0(D) + R_1(D) g(D) \right] + \left[P_0(D) + P_1(D) \right] D^n + \\
&\quad \left[P_0(D) + P_1(D) + P_2(D) + R_1(D) g(D) \right] D^{2n} + \left[P_0(D) \right. \\
&\quad + P_1(D) + P_2(D) + P_3(D) \Big] D^{3n} + \left[P_1(D) + P_2(D) + P_3(D) \right. \\
&\quad + P_4(D) \Big] D^{4n} + \dots + \left[P_{i-3}(D) + P_{i-2}(D) + P_{i-1}(D) \right. \\
&\quad + P_i(D) \Big] D^{in} + \dots .
\end{aligned} \tag{B7}$$

Let $C_i(D)$ be the coefficient of D^{in} . Then

$$C_0(D) + C_1(D) + C_4(D) + C_5(D) + C_8(D) + C_9(D) + \dots = R_1(D) g(D) \tag{B8}$$

and

$$C_2(D) + C_3(D) + C_6(D) + C_7(D) + C_{10}(D) + C_{11}(D) + \dots = R_1(D) g(D). \tag{B9}$$

Therefore

$$\begin{aligned}
w_H \left[C_0(D) \right] + w_H \left[C_1(D) \right] + w_H \left[C_4(D) \right] + w_H \left[C_5(D) \right] + \dots &\geq \\
w_H \left[R_1(D) g(D) \right]
\end{aligned} \tag{B10}$$

and

$$\begin{aligned}
w_H \left[C_2(D) \right] + w_H \left[C_3(D) \right] + w_H \left[C_6(D) \right] + w_H \left[C_7(D) \right] + \dots &\geq \\
w_H \left[R_1(D) g(D) \right]
\end{aligned} \tag{B11}$$

since a sum of the weights of a set of polynomials is always greater than or equal to the weight of the sum of the polynomials. Hence

$$w_H \left[y^{(1)}(D) \right] \geq 2d_g . \tag{B12}$$

Similarly

$$w_H \left[y^{(2)}(D) \right] \geq 2d_h . \quad (B13)$$

This same argument can be employed for all values of k . The number of coefficient equations similar to (B8) and (B9) is always equal to the weight of the polynomial $(D^n - 1)^{k-1}$. Therefore if $L = w_H \left[(D^n - 1)^{k-1} \right]$, then

$$\begin{aligned} w_H \left[y^{(1)}(D) \right] &\geq Ld_g \\ w_H \left[y^{(2)}(D) \right] &\geq Ld_h \end{aligned} \quad (B14)$$

for all values of k . Note that this agrees with equations (B4) when k is a power of 2.

REFERENCES

1. P. Elias, "Coding for Noisy Channels", IRE Convention Record, Part IV, pp. 37-46, 1955.
2. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Trans. Inform. Theory, IT-13, pp. 260-269, 1967.
3. J. M. Wozencraft and B. Reiffen, Sequential Decoding. Cambridge, Mass.: The M.I.T. Press, 1961.
4. J. L. Massey, Threshold Decoding. Cambridge, Mass.: The M.I.T. Press, 1963.
5. J. L. Massey and M. K. Sain, "Codes, Automata, and Continuous Systems: Explicit Interconnections", IEEE Trans. Automatic Control, AC-12, pp. 644-650, 1967.
6. G. D. Forney, "Algebraic Structure of Convolutional Codes", IEEE International Symposium on Information Theory, 1969.
7. J. P. Robinson, "Error Propagation and Definite Decoding of Convolutional Codes", IEEE Trans. Inform. Theory, IT-14, pp. 121-128, 1968.
8. J. L. Massey, "Some Algebraic and Distance Properties of Convolutional Codes", Error Correcting Codes (Ed. H. B. Mann). New York: John Wiley and Sons, Inc., 1968.
9. J. L. Massey, Private Communication, University of Notre Dame, Notre Dame, Indiana, 1967.
10. R. McEliece and H. Rumsey, "Capabilities of Convolutional Codes", Jet Propulsion Laboratory SPS 37-50, Vol. III, 1968.
11. B. Neumann, "Distance Properties of Convolutional Codes", M. S. Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass., August 1968.
12. J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits", IEEE Trans. on Computers, C-17, pp. 330-337, 1968.
13. R. R. Olson, "Note on Feedforward Inverses for Linear Sequential Circuits", Tech. Report EE-684, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, April 1968.

14. J. L. Massey, "Reversible Codes", Inform. Control, Vol. 7, pp. 369-380, 1964.
15. J. P. Robinson, "Reversible Convolutional Codes", IEEE Trans. Inform. Theory, IT-14, pp. 609-610, 1968.
16. J. P. Robinson, "An Upper Bound on the Minimum Distance of a Convolutional Code", IEEE Trans. Inform. Theory, IT-11, pp. 567-571, 1965.
17. R. W. Kolor, "A Gilbert Bound for Convolutional Codes", M. S. Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass., June 1967.
18. T. J. Wagner, "A Gilbert Bound for Periodic Binary Convolutional Codes", IEEE Trans. Inform. Theory, IT-14, pp. 752-755, 1968.
19. E. N. Gilbert, "A Comparison of Signalling Alphabets", Bell System Tech. J., Vol. 31, pp. 504-522, 1952.
20. M. Plotkin, "Binary Codes with Specified Minimum Distances", IEEE Trans. Inform. Theory, IT-6, pp. 445-450, 1960.
21. R. W. Hamming, "Error Detecting and Error Correcting Codes", Bell System Tech. J., Vol. 29, pp. 147-160, 1950.
22. C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels", Inform. Control, Vol. 10, pp. 65-103, 522-552, 1967.
23. T. N. Morrissey, Private Communication, University of Notre Dame, Notre Dame, Indiana, 1968.
24. E. K. Bower and S. J. Dwyer, III, "A Strengthened Asymptotic Gilbert Bound for Convolutional Codes", IEEE Trans. Inform. Theory, IT-15, pp. 433-435, 1969.
25. R. G. Gallager, Information Theory and Reliable Communication. New York: John Wiley and Sons, Inc., 1968.
26. E. A. Bucher, "Error Mechanisms for Convolutional Codes", Ph.D. Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass., September 1968.
27. E. R. Berlekamp, Algebraic Coding Theory. New York: McGraw Hill, Inc., 1968.
28. G. D. Forney, "Final Report on a Study of a Simple Sequential Decoder", U. S. Army Satellite Communication Agency Contract DAAB07-68-C-0093, Appendix A, Codex Corporation, Watertown, Mass., April 1968.

29. S. Lin and H. Lyne, "Some Results on Binary Convolutional Code Generators", IEEE Trans. Inform. Theory, IT-13, pp. 134-139, 1967.
30. J. J. Bussgang, "Some Properties of Binary Convolutional Code Generators", IEEE Trans. Inform. Theory, IT-11, pp. 90-100, 1965.
31. C. E. Shannon, "A Mathematical Theory of Communication", Bell System Tech. J., Vol. 27, pp. 379-423, 623-656, 1948.
32. J. L. Massey, "Convolutional Coding Techniques for Data Protection", Quarterly Progress Report, NASA Grant NGL-15-004-026, University of Notre Dame, Notre Dame, Indiana, June 1969.
33. I. M. Jacobs, "Sequential Decoding for Efficient Communication from Deep Space", IEEE Trans. Communication Technology, COM-15, pp. 492-501, 1967.
34. A. D. Wyner, "Gilbert Bounds for Recurrent Codes", Research Paper, IBM Watson Research Center, Yorktown Heights, New York, July 1963.
35. D. D. Sullivan, "A Class of Double Error Correcting Convolutional Codes Derived from BCH Codes", unpublished memorandum, 1968.
36. D. D. Sullivan, "Control of Error Propagation in Convolutional Codes", Ph.D. Thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, 1966.
37. A. D. Wyner and R. B. Ash, "Analysis of Recurrent Codes", IEEE Trans. Inform. Theory, IT-9, pp. 143-156, 1963.
38. A. D. Wyner, "On the Equivalence of Two Convolution Code Definitions", IEEE Trans. Inform. Theory, IT-11, pp. 600-602, 1965.
39. R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes", Inform. Control, Vol. 3, pp. 68-79, 279-290, 1960.